30-
242402

WAVETEK®
Wavetek Rockland Scientific, Inc.
10 Volvo Drive
Rockleigh, N.J. 07647
(201) 767-7900

PROGRAMMING THE 804A IN SAMPL™

[illegible]

Instruction Manual

for

PROGRAMMING THE 804A
IN SAMPL™

WAVETEK®
Wavetek Rockland Scientific, Inc.
10 Volvo Drive
Rockleigh, N.J. 07647
(201) 767-7900

WARRANTY

All Nicolet Scientific instruments are warranted against defective materials and workmanship for one year. For detailed information on the warranty agreement, refer to the page at the rear of this publication for conditions and shipping instructions.

**Copyright©1982 by Nicolet Scientific Corporation.
All Rights Reserved.**

Printed in the United States of America. All rights reserved. Contents of this publication may not be reproduced in any form without permission of the copyright owner.

PROPRIETARY NOTICE

This information is furnished only for test and maintenance purposes. The data is proprietary in nature and should be treated accordingly. The data shall not be either released or used, duplicated, or disclosed in whole or in part for manufacture or procurement without the written permission of Nicolet Scientific Corporation, except for (i) emergency, repair or overhaul work by or for the government, where the item or process concerned is not otherwise reasonably available to enable timely performance of work; or (ii) release to a foreign government, as the interests of the United States may require; provided that in either case the release, use, duplication or disclosure hereof shall be subject to the foregoing limitations. This legend will be marked on any reproduction hereof in whole or in part.

CONTENTS

SECTION PAGE

PART 1 - PROGRAMMING FOR NON-PROGRAMMERS

Chapter 1 - Introduction

1-1. Programming the 804A 1
1-2. The 804A Programming Language..... 1
1-3. Program Syntax..... 4

Chapter 2 - 804A Math Operations

2-1. 804A Math..... 7
2-2. Reverse Polish Notation..... 8
2-3. Stacks..... 8
2-4. The Scalar Stack Registers..... 9
2-5. Roll, Enter, and Exchange..... 10
2-6. Arithmetic Operations..... 13
2-7. Scalar, Mixed, and Vector Modes..... 16

Chapter 3 - 804A Programming Technique

3-1. Immediate Mode Programming..... 21
3-2. Direct Mode Programming..... 23
3-3. Learn Mode Programming..... 24
3-4. Programming Measurement Setups..... 27

Appendix - A Programming Example

A-1. Program: DBCAL.MC..... 29
A-2. Program: DISTLIST..... 34

PART 2 - THE 804A PROGRAMMER'S HANDBOOK

Chapter 4 - Entry and Notation

4-1. 804A Program Entry..... 45
4-2. Notation of this Handbook..... 45

TABLE OF CONTENTS (CONT'D.)

Chapter 5 - Mnemonics

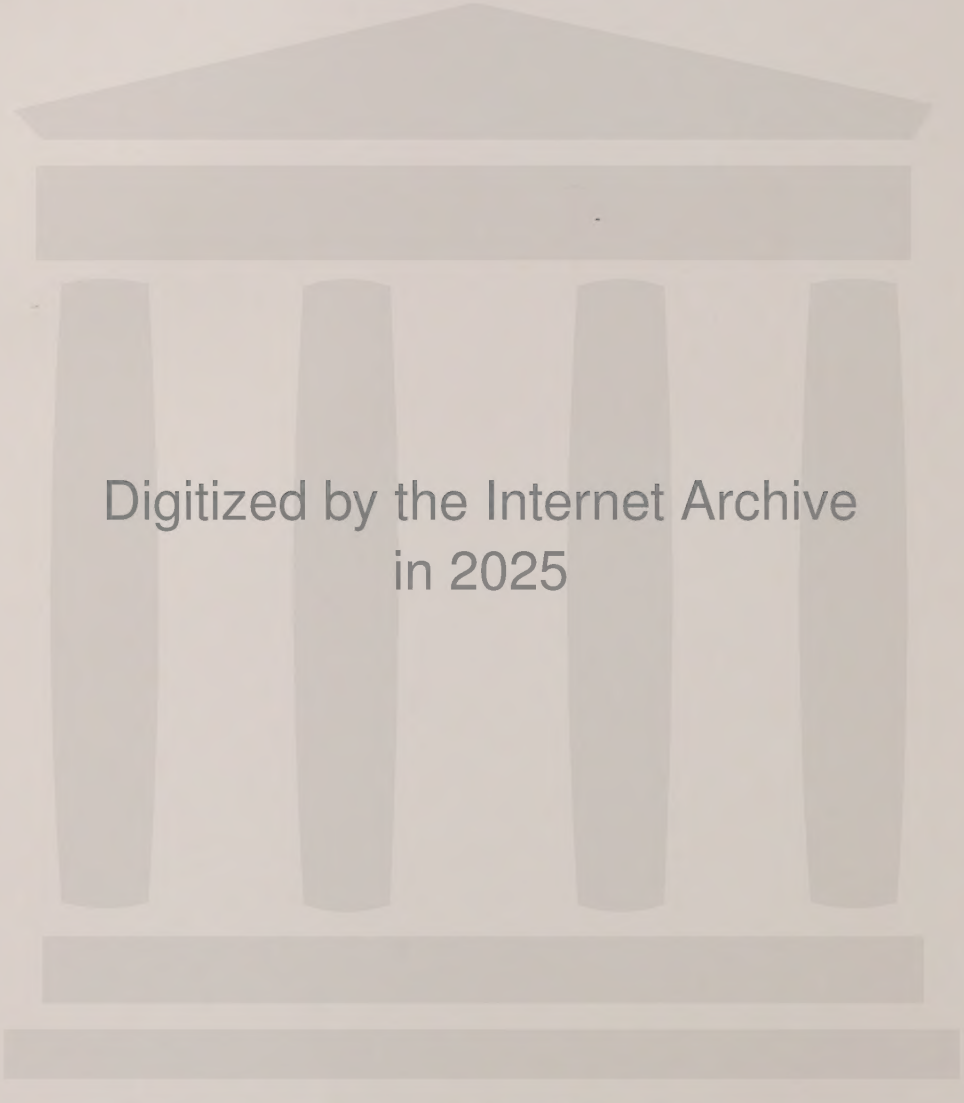
5-1. Basic Operation Mnemonics..... 47
1. Alphabetical Order..... 47
2. Order by Function Group..... 57
A. System Configuration Commands..... 57
B. Averager Control Commands..... 58
C. Display Control Commands..... 59
D. Calibration and Reference Commands..... 62
E. Cursor and Marker Control Commands..... 63
F. Transient Capture & Delay Commands..... 64
G. Input Sensitivity and Coupling..... 65
H. Weighting..... 65
I. Frequency and Order Controls..... 66
J. Disk and Other Storage Controls..... 66
K. Plotter Control Commands..... 68
L. Noise Source Control..... 70
M. Text and Program Entry Control..... 70

Chapter 6 - Extended Mnemonics

6-1. Extended Operation Mnemonics..... 71
1. Alphabetical Listing..... 71
2. Error Codes after "serr"..... 80

Chapter 7- Extended Commands in Functional Order

7-1. Math Mode Control..... 83
7-2. Stack Control..... 83
7-3. Basic Math Operations..... 83
7-4. Extended Math Operartions..... 84
7-5. Transcendental Functions..... 84
7-6. Format Conversion..... 84
7-7. Frequency/Time Transformations..... 84
7-8. Vector Generation..... 84
7-9. Comparison and Peak..... 85
7-10. Cursor Control..... 85
7-11. Access Parameter of Data Memory Associated with the Active Cursor..... 85
7-12. Weighting Function Generation and Recall..... 86
7-13. Conditional, Branching, and Program Control..... 86
7-14. I/O..... 87
7-15. Miscellaneous Commands..... 87


Digitized by the Internet Archive
in 2025

https://archive.org/details/manualsplus_09830

PART 1 CHAPTER 1

INTRODUCTION

1-1. Programming the Model 804A

The Model 804A Multi-Channel Signal Processor is a powerful programmable measurement and data reduction system. In order to make use of its full potential, it is necessary to learn how to write macro-programs which can be executed on the 804A.

A macro-program is a series of control instructions which brings the 804A, in sequence through a desired set of measurement or analysis functions. The 804A macro-programs can branch, that is, change the order of program execution, based on the results of testing the data being measured. The 804A supports subroutines, thereby simplifying the programming operation. Using these features, it is possible to simplify many measurements by reducing a series of control operations into the simple execution of a macro-program. It is also possible to use a macro-program to guide an inexperienced operator by means of prompted semi-automatic or fully automatic operation of the Model 804A.

Part 1 of this manual is intended for 804A users who have had little or no programming experience. The only requisite skill is the ability to visualize the solution to a problem in a clear and logical way.

Experienced programmers will be able to skim over this more basic material and make use of the reference tables in Part 2 of this manual.

1-2. The 804A Programming Language

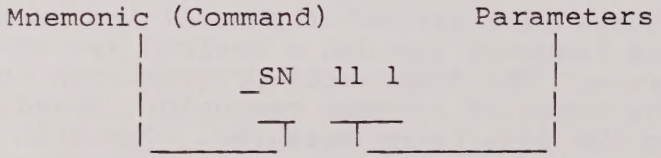
The macro-programming language used in the Model 804A is specifically intended for signal processing applications. The language is called SAMPL™, an acronym which stands for Signal Analysis Macro Programming Language.

There are two types of program commands or mnemonics used in SAMPL. The first, called a basic or control command, is the programmed equivalent of pressing a control key on the primary keyboard. The second type of command is the extended command. Extended commands generally do not correspond to any control key on the primary keyboard. However, any SAMPL command can be executed immediately from the 804A keyboard by typing it into the entry field. This will be

discussed in Section 3-1, "Immediate Mode Programming". Chapter 3, "804A Programming Technique." The same SAMPL commands are used in immediate mode, stored programs, and to control the 804A from either interface.

Basic SAMPL commands consist of two upper-case letters, associated with one or two numerical parameters (arguments), or alpha Labels. All commands must have a leading space which is represented below by the underscore (_). For example:

SELECTING INPUT SENSITIVITY(SN) OF 1 VOLT(11) ON CHANNEL A(1)



There are four possible types of parameters or arguments that are associated with the basic command group. Each type of parameter has an associated symbol to identify it in the tables that appear in Part 2 of this manual. The types of parameters are as follows:

SYMBOL	TYPE	DESCRIPTION
N	NUMBER	Direct entry of a numeric parameter. Example: 256 _AN Set number of averages to 256.
T	TEXT	Entry of filename or text string. Example: "SETUP _DK 2 Recall disk filename called SETUP.
C	CHOICE	Selection of soft key associated with desired operation. C is a number between 1 and 16 corresponding to one of the possible soft key labels for the desired control function. Example: _CT 1 Select single cursor. _CT 2 Select sideband cursor. _CT 3 Select harmonic cursor.

(cont.)

SYMBOL	TYPE	DESCRIPTION
S	SELECTOR	Select desired input channel, processing memory, or display trace. S is a number between 1 and 4.

Example: _IT 1 1 Select AC coupling on input channel 1.
_IT 1 4 Select AC coupling on input channel 4.

The general forms of command syntax are

1. N
   _XX S

2. T
   _XX S

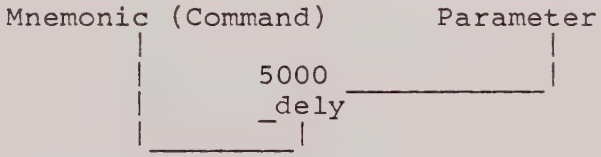
3.
   _XX C S

where XX represents an arbitrary command. The characters N, T, C, and S are explained in the above chart. In some commands, the S parameter may not be applicable and is left off. Note that there must be a space, represented above by the underscore character (_), preceding each command or mnemonic. Note also that the first character in a text string appearing independently on a line must be a quotation mark (").

The second type of command is the extended command or mnemonic. Extended commands are used to program math, data measurements, and program control functions. The extended command consists of three or four lower case letters and, like the basic commands, may include alpha text strings or numeric parameters. Extended commands must also be preceded by a leading space. Some extended commands require numeric data to be stored in the scalar math stack registers. (This is discussed in more detail in Chapter 2, "804A Math Operations.")

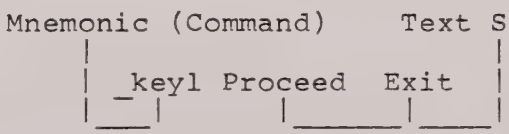
For example:

DELAY PROGRAM EXECUTION (dely) FOR 5 SECONDS (5000 msec)



OR

LABEL SOFTKEYS (keyl) WITH THE LABELS "PROCEED" AND "EXIT"



The general forms of extended command syntax are

1. _xxxx 2. N _xxxx 3. _xxxx T T T T...

where xxxx is an arbitrary extended command, N is a number, and T is a text string.

All commands must be preceded by a space, represented above by the underscore (_). Some commands permit multiple text strings. Text strings following a command do not require a quotation mark (").

Complete information about each command can be found in Chapter 5. The descriptions contain all the information regarding mnemonic symbols, labels, and parameters. The mini-manual, in a plastic sleeve located to the right of the disk drive, and the help file, resident in the 804A provide concise, readily available descriptions of each command.

1-3. Program Syntax

Program syntax, is a system of rules for organizing the elements of a program. The syntax of SAMPL is very simple. There are five rules.

1. All commands (mnemonics) must be preceded by a space.

2. All numeric entries without a leading space are interpreted as parameters or operands. They are automatically moved to the entry field.

3. Alpha-numeric text strings such as program or disk file names which must be entered via the entry field, must be preceded by a leading quotation mark, ".

4. Program labels must end in a colon, :.

5. Comments must be preceded by an asterisk, *.

Here is an example of a program illustrating these rules.

```
ATAN: <--- Colon indicates that this is a label.

* Program to calculate arctan IM(x)/RE(x)
|
|__________Asterisk indicates a comment.

"ATAN.SU
|
|__________Quotation mark indicates filename to entry field.
           This is required by disk recall which follows.

 DK 2 <--- Uppercase indicates basic command mnemonic.
|
|__________Leading space indicates a command.


 recl X <- Lowercase indicates extended command mnemonic.
|
|__________ Leading space indicates a command.

* IMAGINARY PART OF LN X=jARCTAN(IMAG(X)/REAL(X))

 lng

* EXTRACT IMAG(LN X)

 imag

* CONVERT TO REAL NUMBER IN UNITS OF DEGREES

0,-57.29578
|
|_________Operand entered in complex form (Real, Imaginary).

 tims

 stop
```

CHAPTER 2

804A MATH OPERATIONS

2-1. 804A Math

Before proceeding with more detailed instruction on programming, it is necessary to provide some information on 804A math operations.

The post-processing mathematics capability of the 804A is specifically designed to handle signal processing problems. It incorporates the ability to manipulate single complex numbers (scalar mode) and multiple complex number arrays (vector mode), as well as combinations of both (mixed mode). The possible mathematics operations include

1. Basic Arithmetic:
   Addition (+), Subtraction (-), Multiplication (*), Division (/)

2. Non-linear Functions:
   Reciprocals, Squares, Roots

3. Exponentials and Logarithms:
   e^x, 10^x, ln, log

4. Pseudo-Integration, Differentiation:
   *jω, /jω

5. Complex Conjugate:
   X'

6. Transcendental Functions:
   Cosine, Sine

7. Vector Generation:
   Box, Notch, Ramp, Constant

8. Data Comparison and Peak

9. Test Conditions, Branching, and Program Control

10. Time/Frequency Transformations:
    FFT, IFFT

11. Access Parameters and Data Measurement

2-2. Reverse Polish Notation

The 804A mathematics uses reverse Polish notation (RPN). In RPN, the operator is written after the operand, as in XY+, instead of in between the operands, as in X+Y. RPN is also called postfix because the operator is after the operands. The more common notation is called infix. The name RPN is used in recognition of the Polish logician, J. Lukasiewicz, who studied this notation.

RPN has several advantages over infix for expressing algebraic formulas. First, any formula can be expressed without parentheses. Second, and related to the above point, RPN makes natural and efficient use of a data structure called a "stack," which will be introduced in Section 2-3. This saves keystrokes and execution time. Third, RPN does not arbitrarily assign precedence to operators as normal algebraic notation does.

As an example of the advantages of RPN, consider the simple algebraic formula a+b*c. It must be recognized that b*c is done before the addition because multiplication has precedence over addition. Thus, execution does not take place left to right. Executing this formula on an algebraic calculator would require the keystrokes a, +, (, b, *, c,), =; a total of eight keystrokes. Executing this formula on an RPN calculator would require the keystrokes a, enter, b, enter, c, *, +; a total of seven keystrokes.

It is very easy to visualize how RPN works once the concept of a stack is understood. The 804A's two stacks are explained in Section 2-3.

2-3. Stacks

All data, whether scalar or vector, is stored and manipulated in memory storage registers, called "stack" registers. This name is a result of the organization of the registers. They may be visualized as being "stacked" one on top of the other.

There are four registers for each data type. Scalar stack registers are labeled x, y, z, and t. Vector stack registers are called X, Y, Z, and T. The main difference between the scalar and vector stacks is the capacity of each register. Scalar registers hold one complex number (both real and imaginary parts). The vector registers can hold up to 513 complex numbers or 1,024 real numbers, representing a data array such as a time history or a frequency spectrum. The diagrams in Section 2-4 will help illustrate the stack concept.

2-4. Scalar Stack Registers

Stack Register Name		Stack Contents REAL , IMAGINARY	
	t	0.0 , 0.0	
	z	0.0 , 0.0	
	y	0.0 , 0.0	
Entry Field-->	x	0.0 , 0.0	<--

The x register is visible to the user as the entry field which appears in the upper right hand corner of all the 804A displays. Notice that the scalar stack is set up to store complex numbers, i.e., numbers having a real and an imaginary part.

Numbers are entered into the entry field using the numeric keypad on the primary keyboard. Entries may be real, imaginary, or complex. The format for numeric entry is REAL,IMAGINARY. For example,

1.	Real	1.2, 0 346785.99 -468
2.	Imaginary	0, 322 0, 1.45 0, 3.2E-03
3.	Complex	1, 1 1.3E+03, 1.7E+02 24.3, 1

Math mode is entered by pressing the MATH shift key on the primary keyboard. The scalar math type is then selected using the SCALAR/VECTOR softkey (the eighth softkey).

Entering a number into the entry field and pressing the SET key on the numeric keypad overwrites the current contents of the x scalar stack register, for example:

Type 2, 2.6

The stack now looks like this:

Stack Register---↓		Stack Contents REAL	, IMAGINARY	
	t	0.0	, 0.0	
	z	0.0	, 0.0	
	y	0.0	, 0.0	
Entry Field -->	x	2.0	, 2.6	<--

Note that the number 2,2.6 has been entered into the x scalar stack register which is visible as the entry field.

2-5. Roll, Enter, and Exchange

It is possible to look at the contents of the stack in the scalar math mode by pressing the ROLL softkey. Each time the ROLL key is depressed, the stack contents are shifted downward by one register. It is, therefore, possible to shift the contents of any register down to the x register by one or more ROLL operations.

	BEFORE REAL	, IMAGINARY
t	4.0	, 4.0
z	3.0	, 3.0
y	2.0	, 2.0
x	1.0	, 1.0

ROLL

	AFTER REAL	, IMAGINARY
t	1.0	, 1.0
z	4.0	, 4.0
y	3.0	, 3.0
x	2.0	, 2.0

The example above illustrates the effect of the ROLL operation. The previous contents of the y stack register are now in x and are, therefore, visible. Pressing the ROLL softkey again will bring the original contents of the z stack register into the x register where it may be examined.

	BEFORE REAL	, IMAGINARY
t	1.0	, 1.0
z	4.0	, 4.0
y	3.0	, 3.0
x	2.0	, 2.0

ROLL

	AFTER REAL	, IMAGINARY
t	2.0	, 2.0
z	1.0	, 1.0
y	4.0	, 4.0
x	3.0	, 3.0

Another ROLL operation will move the original contents of the t stack register into view.

	BEFORE		ROLL		AFTER	
	REAL	IMAGINARY			REAL	IMAGINARY
t	2.0	2.0		t	3.0	3.0
z	1.0	1.0		z	2.0	3.0
y	4.0	4.0		y	1.0	1.0
x	3.0	3.0		x	4.0	4.0

By depressing the ROLL softkey four times the user can review the entire contents of the stack and return the stack contents to their original locations.

	BEFORE		ROLL		AFTER	
	REAL	IMAGINARY			REAL	IMAGINARY
t	3.0	3.0		t	4.0	4.0
z	2.0	2.0		z	3.0	3.0
y	1.0	1.0		y	2.0	2.0
x	3.0	3.0		x	4.0	4.0

The ENTER softkey is used to put numbers into the stack registers. A number, placed into the x stack register (entry field) by use of the numeric keypad, can be copied into the y stack register by pressing the ENTER softkey.

	BEFORE		ENTER		AFTER	
	REAL	IMAGINARY			REAL	IMAGINARY
t	0.0	0.0		t	0.0	0.0
z	0.0	0.0		z	0.0	0.0
y	0.0	0.0		y	2.0	0.0
x	2.0	0.0		x	2.0	0.0

The number 2.0, entered by use of the keypad into the entry field (x stack register), is copied into the y stack register. The ENTER operation also causes the contents of the y and z stack registers to be transferred upward in the stack to the z and t, respectively. The contents of the t register are lost.

Note:

Enter and clr X disable the automatic stack lift. All other operations that use the stack enable automatic stack lift. Operations that do not affect the stack (such as basic 804A operations) do not affect automatic stack lift. When automatic stack lift is enabled, placing new data into the x stack register causes an enter to be done first.

As a short example of automatic stack lift, consider the operation of sin(a)+b. First, the value "a" is typed in, and then the sin function is executed. The value "b" is then typed in. When "b" is typed in, the value of sin(a) is automatically copied into y by the automatic stack lift. Now "+" will produce sin(a)+b. Automatic stack lift works on the scalar and vector stacks. The extended functions ramp, box, ntc, and cnst (explained below), for example, all create new data in the X vector stack register. They are treated as numeric entry; that is, they will cause an automatic enter if stack lift is enabled.

Now, to continue the example of the ENTER operation, the action of two more ENTERs is illustrated below.

	BEFORE	
	REAL	IMAGINARY
t	0.0	0.0
z	0.0	0.0
y	2.0	0.0
x	2.0	0.0

ENTER

	AFTER	
	REAL	IMAGINARY
t	0.0	0.0
z	2.0	0.0
y	2.0	0.0
x	2.0	0.0

	BEFORE	
	REAL	IMAGINARY
t	0.0	0.0
z	2.0	0.0
y	2.0	0.0
x	2.0	0.0

ENTER

	AFTER	
	REAL	IMAGINARY
t	2.0	0.0
z	2.0	0.0
y	2.0	0.0
x	2.0	0.0

The remaining stack operation is EXC which stands for exchange. The EXC operation will interchange the contents of the x and y stack registers. This enables the reordering of the numbers or data in the stacks.

	BEFORE	
	REAL	IMAGINARY
t	0.0	0.0
z	0.0	0.0
y	1.0	0.0
x	2.0	0.0

EXC

	AFTER	
	REAL	IMAGINARY
t	0.0	0.0
z	0.0	0.0
y	2.0	0.0
x	1.0	0.0

	BEFORE	
	REAL	IMAGINARY
t	0.0	0.0
z	0.0	0.0
y	2.0	0.0
x	1.0	0.0

EXC

	AFTER	
	REAL	IMAGINARY
t	0.0	0.0
z	0.0	0.0
y	1.0	0.0
x	2.0	0.0

2-6. Arithmetic Operations

Arithmetic operations are performed by first entering the numbers in the stack, positioning them just as if on a piece of paper. Once the numbers are entered, the desired operation is performed. For example:

```
Add :  2
      +3
      ---
       ?
```

Using the primary keyboard-

1. Recall math softkeys to display by pressing MATH key.
2. Select scalar mode using scalar/vector soft key.
3. Using the numeric keypad, type 2 into the entry field.
4. Press the enter softkey.
5. Type in the number 3.
6. Press the + softkey.
7. Read sum in the entry field.

Or, using extended mnemonics from the optional secondary keyboard-

1. Type sscl and press return.
2. Type 2 into the entry field.
3. Press return (the 2 overwrites the contents of x).
4. Type ent and press return.
5. Type in the number 3.
6. Type pls and press return.
7. Read sum in the entry field.

In a program, this would look like:

```
 sscl
2
 ent
3
 pls
```

Here is what the stack register contents look like before and after each operation. The XX, YY, ZZ, TT indicate arbitrary numbers which exist in the stack as a result of previous operations.

	BEFORE			+		AFTER	
	REAL	IMAGINARY				REAL	IMAGINARY
t	ZZ	ZZ			t	ZZ	ZZ
z	YY	YY			z	ZZ	ZZ
y	2.0	0.0			y	YY	YY
x	3.0	0.0			x	5.0	0.0

Notice that the + operation caused the contents of the x and y stack registers to be added and the result to be placed into the x stack register. Although it was not obvious in the example, when the + operation was performed the contents of each of the remaining stack registers (i.e., z and t) were moved down to the next lower register. Since t is the highest register in the stack, it retains the original data.

Automatic raising and lowering of the contents in the stack registers makes RPN easy and efficient to use, especially when dealing with chained operations. The stack contents are raised if a number is typed in after an operation. The stack contents are automatically lowered when operations involving both x and y stack registers are performed. For example:

(17+27) / (65-54)=?

	BEFORE			17		AFTER	
	REAL	IMAGINARY				REAL	IMAGINARY
t	TT	TT			t	TT	TT
z	ZZ	ZZ			z	ZZ	ZZ
y	YY	YY			y	YY	YY
x	XX	XX			x	17	0.0

BEFORE

	REAL	IMAGINARY
t	TT	TT
z	ZZ	ZZ
y	YY	YY
x	17.0	0.0

ENTER

AFTER

	REAL	IMAGINARY
t	ZZ	ZZ
z	YY	YY
y	17.0	0.0
x	17.0	0.0

BEFORE

	REAL	IMAGINARY
t	ZZ	ZZ
z	YY	YY
y	17.0	0.0
x	17.0	0.0

27

AFTER

	REAL	IMAGINARY
t	ZZ	ZZ
z	YY	YY
y	17.0	0.0
x	27.0	0.0

BEFORE

	REAL	IMAGINARY
t	ZZ	ZZ
z	YY	YY
y	17.0	0.0
x	27.0	0.0

+

AFTER

	REAL	IMAGINARY
t	ZZ	ZZ
z	ZZ	ZZ
y	YY	YY
x	44.0	0.0

BEFORE

	REAL	IMAGINARY
t	ZZ	ZZ
z	ZZ	ZZ
y	YY	YY
x	44.0	00

65

AFTER

	REAL	IMAGINARY
t	ZZ	ZZ
z	YY	YY
y	44.0	0.0
x	65.0	0.0

BEFORE

	REAL	IMAGINARY
t	ZZ	ZZ
z	YY	YY
y	44.0	0.0
x	65.0	0.0

ENTER

AFTER

	REAL	IMAGINARY
t	YY	YY
z	44.0	00
y	65.0	0.0
x	65.0	0.0

BEFORE

	REAL	IMAGINARY
t	YY	YY
z	44.0	0.0
y	65.0	0.0
x	65.0	0.0

54

AFTER

	REAL	IMAGINARY
t	YY	YY
z	44.0	0.0
y	65.0	0.0
x	54.0	0.0

BEFORE

	REAL	IMAGINARY
t	YY	YY
z	44.0	0.0
y	65.0	0.0
x	54.0	0.0

-

AFTER

	REAL	IMAGINARY
t	YY	YY
z	YY	YY
y	44.0	0.0
x	11.0	0.0

BEFORE

	REAL	IMAGINARY
t	YY	YY
z	YY	YY
y	44.0	0.0
x	11.0	0.0

/

AFTER

	REAL	IMAGINARY
t	YY	YY
z	YY	YY
y	YY	YY
x	4.0	0.0

The automatic stack movement allows the retaining and repositioning of intermediate results without having to reenter the numbers. It also makes it possible to enter numbers in any logical order. These features make the most complex problems much easier to handle.

2-7. Scalar, Mixed, and Vector Modes

Although stack operations have only been discussed in the scalar math mode, they are identical in the vector mode. Any math operation or function may be applied to either vector or scalar data. The only limitation in vector operations is that the contents of vector stack registers involved in a math operation are of the same data type (either time or frequency). The vector mode stack registers are the X, Y, Z, and T processor memories. This will permit data gathering and post-processing within the same memory. Note that when any vector math operation is performed, the update status of the affected vector stack register (processor memory) is changed from LIVE to FREEZE. This will prevent the accidental loss of data due to inadvertent overwriting of new data over post-processed data. Vector mode is entered by pressing the alternate action SCALAR/VECTOR softkey a number of times until only label VECTOR is displayed in reverse video.

The mixed mode permits the contents of the x scalar stack register to be used as a constant in order to operate on the X vector stack register by using the four arithmetic functions +, -, *, and /. All other math operations in the mixed mode are identical to the vector mode. Mixed mode is entered by pressing the SCALAR/VECTOR softkey a number of times until both labels are displayed in inverse video.

Note:

The extended commands "svec" and "smix" can be used to set vector and mixed modes directly. The extended command "sscl", already introduced, sets scalar mode.

Math functions, such as log, sine, exponentiation, etc., operate on either the x scalar stack register or the X vector stack register depending on the mode selected. In either mode, the argument of the function must be in the appropriate stack register. For example:

$12^2 = ?$

BEFORE

	REAL	IMAGINARY
t	TT	TT
z	ZZ	ZZ
y	YY	YY
x	XX	XX

12

AFTER

	REAL	IMAGINARY
t	TT	TT
z	ZZ	ZZ
y	YY	YY
x	12.0	0.0

BEFORE

	REAL	IMAGINARY
t	TT	TT
z	ZZ	ZZ
y	YY	YY
x	12.0	0.0

x^2

AFTER

	REAL	IMAGINARY
t	TT	TT
z	ZZ	ZZ
y	YY	YY
x	144.0	0.0

The next numeric entry will cause the stack to automatically ENTER the result, lifting it into the y stack register.

Some extended commands in programming require that numerical constants, such as frequency limits, be entered in the x and y scalar stack registers. This can be accomplished by using the ENTER and EXCNG operations. In some of these operations, the order in which the numbers are entered is significant. For example:

_puty ; This extended command places the data value stored in the y scalar stack register at the horizontal cell of the X vector stack register specified by the number stored in the x scalar stack register.

Data Entry:

1. Use the math shift key to bring math softkeys up on the display.

2. Select scalar mode using the SCALAR/VECTOR softkey.

3. Type in the desired x value.

4. Press the enter softkey.

5. Type in the desired y value.

6. Press the EXCNG softkey.

7. Type puty.

8. Press the RETURN key to execute command.

This is what the operations outlined above did in the stack registers.

X VALUE

	BEFORE	
	REAL	IMAGINARY
t	TT	TT
z	ZZ	ZZ
y	YY	YY
x	XX	XX

	AFTER	
	REAL	IMAGINARY
t	TT	TT
z	ZZ	ZZ
y	YY	YY
x	x value,	0.0

ENTER

	BEFORE	
	REAL	IMAGINARY
t	TT	TT
z	ZZ	ZZ
y	YY	YY
x	x value,	0.0

	AFTER	
	REAL	IMAGINARY
t	ZZ	ZZ
z	YY	YY
y	x value,	0.0
x	x value,	0.0

Y VALUE

	BEFORE	
	REAL	IMAGINARY
t	ZZ	ZZ
z	YY	YY
y	x value,	0.0
x	x value,	0.0

	AFTER	
	REAL	IMAGINARY
t	ZZ	ZZ
z	YY	YY
y	x value,	0.0
x	y value,	0.0

	BEFORE		EXCNG		AFTER	
	REAL	IMAGINARY			REAL	IMAGINARY
t	ZZ	ZZ		t	ZZ	ZZ
z	YY	YY		z	YY	YY
y	xvalue	0.0		y	yvalue	0.0
x	yvalue	0.0		x	xvalue	0.0

Note that the EXCNG operation changed the order of the data in the x and y stack registers without affecting the z and t stack registers.

This same operation could have been handled in a macro-program as follows:

```
* PROGRAM TO DEMONSTRATE PARAMETER ENTRY x=500,y=1.2
500
 ent
1.2
 exc
 puty
```

The ability to manipulate data efficiently comes with practice. For example, the above program could be one line shorter:

```
* PROGRAM TO DEMONSTRATE PARAMETER ENTRY x=500,y=1.2
1.2
 ent
500
 puty
```

Of course, these operations can also be performed from the primary keyboard (using the math and alpha shifts) or from the secondary keyboard.

Additional information on post-processing math is available in the Model 804A instuction manual. After reading this section, the user should be able to enter numbers into the scalar stack registers. This ability will be required in the following sections devoted to SAMPL programming.

CHAPTER 3

804A PROGRAMMING TECHNIQUE

3-1. Immediate Mode Programming

The macro-programming commands, which have been discussed, can be used to control the 804A in a variety of ways. The simplest programming technique is the "immediate mode." Immediate mode programming is used manually to execute single-line basic or extended commands. This is the only method of manually executing some of the extended commands which have no equivalent keyboard or softkey control.

Immediate mode programming is performed by typing the command or mnemonic into the entry field and then depressing the RETURN key. Note that if the optional keyboard is not used, the equivalent function is the ALPHA shift followed by pressing the LINE key. If a command requires a preceding numeric parameter, that parameter must be entered into the x scalar stack register. This can be accomplished by typing the number into the entry field and pressing the RETURN key (or ALPHA shift followed by LINE) prior to entering the mnemonic. Commands requiring text strings on the same line as the mnemonic must fit within the 22 character entry field. Commands which are too long or which contain preceding text cannot be executed in immediate mode.

EXAMPLES:

Inverse Fourier Transform data in X vector stack.
Inverse Fourier Transform command IFFT.

1. Press ALPHA key and select lower case.
2. Type ifft.
3. Press LINE key (the IFFT is performed).
4. Press ALPHA key (turns off alpha shift).

Using the optional keyboard, this can be simplified.

1. Type ifft.
2. Press RETURN.

Set frequency coverage span to 20 KHz using the frequency span coverage command (20 KHz) SP 14.

1. Press ALPHA key.
2. Type SP 14.
3. Press LINE key.
4. Press ALPHA key.

Using the optional keyboard-

1. Type SP 14.
2. Press RETURN.

Get location and amplitude of peak between 1,000 and 2,000 Hz using the PEAK command.

1. Press MATH key.
2. Select SCALAR mode using SCALAR/VECTOR softkey.
3. Type 1000.
4. Press ENTER softkey.
5. Type 2000.
6. Press SET.
7. Press ALPHA key and select lower case.
8. Type peak.
9. Press LINE key.
10. Press ALPHA key.
11. x stack register contains frequency of peak.
    y stack register contains amplitude of peak.

Using only the optional keyboard-

1. Type sscl and press RETURN (set scalar mode).

2. Type 1000 and press RETURN.

3. Type ent and press RETURN.

4. Type 2000 and press RETURN.

5. Type peak and press RETURN.

6. x stack register contains frequency of peak.
   y stack register contains amplitude of peak.

3-2. Direct Mode Programming

Direct mode is the most general programming technique. Programs are entered directly into the program buffer area using the text editing features of the Model 804A.

The program buffer, in which the program is stored, can be accessed by pressing the TEXT key once or twice until the full text page is displayed. Note that although it is possible to enter and edit programs in the TEXT/SINGLE display mode, it is preferable to use the 21 line full text page. When the full text page is displayed, select the program buffer by pressing the PROG softkey. If there is no program stored, the display will be blank except for the entry field and the CHARACTERS USED field. The combined capacity of the text/program/auxiliary buffer area is 10,000 characters. This is not a particularly burdensome limit since programs may be chained; that is, one program may recall another from the disk. The CHARACTERS USED field will report the currently available capacity of the buffer in use.

Programs may be entered by first pressing the EDIT softkey and then entering the program mnemonics (using the ALPHA mode on the primary keyboard) or by using the optional keyboard. All program entries must follow the rules of program command syntax discussed earlier. The letters EDT will appear above the entry field to indicate that the 804A is in the EDIT mode.

The 804A can be returned to normal operation by pressing the LIST softkey. This will return the unit to the TEXT display mode. The desired display mode can now be entered by selecting the appropriate DISPLAY key on the primary keyboard.

Once entered into the program buffer, the program can be run by entering the program name or label into the entry field and by pressing the RUN key. A program can also be run by clearing the entry field and pressing the RUN key. In the latter case, the program existing in the program buffer will be run from the first line, regardless of label. The use of program names and labels allows multiple programs or program entry points to be accessed without having to reload programs. When a program is running, the letters RUN will appear above the entry field.

Program execution may be stopped at any time by pressing the STOP key. Once stopped, program execution can be continued from the point at which it stopped by pressing the CONT key.

A program may be executed one command step at a time by single stepping through the program. This feature is helpful in finding faults in a program, a process called debugging. Single step mode can be entered by first pressing the MATH key, and then pressing the CONT key, which is relabeled STEP when in the math shift mode. In step mode the command currently being executed will appear in the lower lefthand corner of the display. In step mode the letters STP will appear above the entry field. If viewing a TEXT or TEXT/SINGLE display the current status of the scalar stack will also be displayed. Note that some commands are not executed in step mode. The Time/Event dependent commands such as "wait" and "dely" will be treated as non operative (no op) commands and control will simply proceed to the next command.

Pressing the STOP key while in the math shift mode will automatically LIST the program in the full text page display mode. Exit from this mode by selecting the desired DISPLAY key.

3-3. Learn Mode Programming

The LEARN mode is a special programming mode intended to simplify program entry. The learn mode is entered by pressing the Math key, to get into math shift mode, and then pressing the RUN key. When in the learn mode, the letters LRN will appear in the area above the entry field on the display. Once in the learn mode, the MATH key may be depressed to exit math shift mode. All subsequent control operations (with the exception of display selection and menu or data cursor movement) will be encoded in SAMPL mnemonics and be sequentially entered into the program buffer area. In this manner, approximately 60 to 70% of an operational program can be written by simply using the 804A.

Pressing the MATH key, returning to math shift mode, and then pressing the RUN key will terminate the learn operation and return the 804A into normal operation. It is recommended that the MATH key be pressed again to exit math shift mode. Display page selections, comments, and prompts can be added to the program at this point by direct programming in the EDIT mode.

The "learned" program may be run, single stepped, or listed just like a direct entry program using the same operations described earlier.

1. Clearing the Program Buffer

The program buffer area may be cleared by using the extended command, clr prog. Exit the EDIT mode and type the mnemonic, clr prog, into the entry field. Press the RETURN key or use the ALPHA shift and LINE keys on the primary keyboard.

2. Storing and Recalling Programs

Programs can be stored and recalled from the internal mini-floppy disk. The operations involved are very simple, they are as follows:

A. Storing

1) The program should be resident in the program buffer.

2) Verify that the DISK STORE selection is PROG (program). (If the program is viewed when STORE is pressed, the DISK STORE selection can also be VIEW.) DISK STORE selection is menu line 24 on the AUX menu page.

3) Type desired filename into entry field. Limit is eight characters plus a two character extension.

4) Press DISK key.

5) Press STORE softkey. It is not necessary to wait for the drive to stop, but the STORE field will be inverted (dark characters on light background) while the store is in progress.

This completes the program store operation. The user may now press the LIST softkey to verify filename, file length, and program data type.

B. Recalling a Program

1) Type the filename of the program into the entry field.

2) Press the DISK key.

3) Press the RECALL softkey.

4) Program has been transferred and is resident in the buffer.

3. Updating and Deleting a Program

A. Updating

1) Modify the program as desired using the TEXT EDIT mode.

2) Exit EDIT mode by pressing the LIST softkey.

3) Type in filename of the program to be updated.

4) Press the DISK key.

5) Press the REPLACE softkey.

6) Press the LIST softkey. Verify filename, program, file length, and program data type. This completes the update operation.

B. Deleting a Program File from the Disk

1) Type program filename into entry field.

2) Press DISK key.

3) Press DELETE softkey.

4) When prompt, "DO YOU WISH TO DELETE", appears, press YES softkey.

5) Wait for disk drive to stop. This completes the procedure.

3-4. Programming Measurement Setups

The Model 804A is a very flexible measurement and processing system. It may require several minutes to choose and set up the desired control states for a specific measurement. Once those states are set up, it is adviseable to save the measurement setup for future use.

There is a SETUP MEMORY available for short term storage. This memory can be accessed via menu page 3, the auxiliary setup menu. The setup may be stored on disk and recalled, either manually or under program control, for later use. (The type of display desired must be selected independently after the setup is recalled.) The following procedures illustrate how this can be accomplished:

1. Saving a Setup (Menu) File

   A. Set up the 804A to make the desired measurement.

   B. Press the CLEAR key to clear the entry field.

   C. Type 24 into the entry field and press LINE.

   D. Press the MENU softkey, then press the ALL softkey. The view line should indicate DISK STORE:MENU ALL.

   E. Type the desired filename, under which the setup will be saved, into the entry field. The filename can have up to eight characters plus a two character extension after a period. Example: TRANSFER.SU.

   F. Press the DISK key.

   G. Press the STORE softkey.

   H. Press the LIST softkey. Verify filename and MENU ALL data type.

The setup is now stored on disk under the filename entered in the entry field.

2. Manual Recall of a Setup File and Display Page

   A. Type filename of setup into the entry field.

   B. Press the DISK key.

   C. Press the RECALL softkey.

   D. Setup is transferred and resident in the 804A.

   E. Press appropriate DISPLAY key for desired display.

3. Programmed Recall of a Setup File and Display Page

The following program steps recall a measurement setup. The setup must be recalled prior to any measurement operation requiring that setup.

```
"FILENAME.SU ; This is the setup filename.
 DK 2        ; Basic command for disk recall.
4            ; Display page selection (see below).
 page        ; Extended command to select display.
```

4. Display Pages

The following numbers are used to identify the display pages for selection under program control.

PAGE NUMBER	DISPLAY TYPE
1	Measurement setup menu.
2	Display setup menu.
3	Auxiliary setup menu.
4	Single display.
5	Dual split display.
6	Dual overlap display.
7	Dual nyquist display.
8	Quad split display.
9	Quad overlap display.
10	Quad split nyquist display.
11	Quad overlap nyquist.
12	Text/Single display.
13	Full page text display.

Please see the "Display Control Commands" section of the 804A Programmer's Handbook (page 57) for more commands that affect the display.

APPENDIX A
(to Part 1)

A PROGRAMMING EXAMPLE

The following program is a typical application of macro-programs. The program involves recalling a setup, measuring data, prompting, math, and scaling display parameters. A detailed discussion of this macro-program will follow. The reader is advised to look up the commands that will be discussed in the tables that appear in Part 2 of this manual.

The program name is DBCAL.MC. It is used to recalibrate the vertical display scales in units of sound pressure level (dB relative to 20 micro pascals). The calibration is based on the output of a microphone which is being excited by an acoustical signal of known sound pressure level. Here is a listing of the program as it would appear in the program buffer of the 804A.

A-1. Program: DBCAL.MC

```
DBCAL.MC:
* Program to calibrate cursor in dB relative to 20 uPa SPL
"DBCAL.SU
 DK 2
4
 page
 prmt APPLY CALIBRATION SOURCE TO A INPUT and press "CONT"
 stop
6
 line
 mcrt
"SELECT SPAN & 'CONT'
 stop
"
 dcon
 VU 10 1
 prmt I AM AUTOSCALING INPUT SENSITIVITY
 SN 8 1
 prmt I AM AUTOSCALING DISPLAY FULL SCALE
 FV 4 1
0
 page
 prmt LOCATE CAL SIGNAL USING CURSOR and press "CONT"
 stop
 curv
 log
```

```
20
 tims
 prmt CALIBRATION, IN DB/1 VOLT, IN ENTRY FIELD; press "CONT"
 stop
 prmt ENTER CALIBRATION SPL (IN DB) and press "CONT"
 stop
 mins
20
 div
 pwr
 CL 1
"20uP"
 UN 1
 prmt RECALIBRATING DISPLAY AND CURSOR SCALING
0
 page
 FV 4 1
 FV 1 1
0
 page
 prmt
" READY
 stop
 goto DBCAL.MC
```

This macro-program can be broken down into a number of simpler blocks for discussion.

```
DBCAL.MC:
```

This is the program name or label. This can be used as an entry point in program execution. Macro-programs can have multiple entry points.

```
* Program to calibrate cursor in dB relative to 20 uPa SPL
```

A leading asterisk indicates that this is a comment.

```
"DBCAL.SU
 DK 2
4
 page
```

This is a disk operation similar to an earlier example. The disk file with the filename DBCAL.SU is transferred from the disk to the 804A. The extension .SU was used to identify this file as a setup file. The data in this file will be stored as setup menus.

The page command selects the single display.

```
prmt APPLY CALIBRATION SOURCE TO A INPUT and press "CONT"
stop
```

The prompt command (prmt) puts the text string, which follows it, on the view line of the display. This text will remain on the screen until the display is updated. The stop that follows the prompt allows time for the operator to see the text and acknowledge it by pressing the CONT key.

```
6
 line
 mcrt
"SELECT SPAN & 'CONT'
 stop
```

This group of commands brings menu line 6, the frequency controls, onto the display. The menu cursor is moved right to allow selection of the frequency span. This will permit the operator to select the appropriate span from among the softkey choices shown on the display.

With a leading quotation mark, the text string is entered into the entry field as an operator prompt. Notice that this text string has fewer than 22 characters, the capacity of the entry field. Again, the stop command allows the user to acknowledge the prompt. The "null" text string, a quotation mark with no subsequent text, is used to clear the entry field after the "prompt" is acknowledged.

```
dcon
VU 10 1
```

These commands turn the data cursor on (dcon) and set the vertical units (VU) on display trace 1 (1) to logarithmic relative (10). These are initialization functions.

```
prmt I AM AUTOSCALING INPUT SENSITIVITY
SN 8 1
```

The prompt describes the command which follows, namely, the autoscaling (8) of the input sensitivity (SN) on input channel A (1). Since there is no action required on the part of the operator, and the inherent delay of the autoscaling operation permits the operator sufficient time to read the text, there is no stop command after the prompt.

```
prmt I AM AUTOSCALING DISPLAY FULL SCALE
FV 4 1
```

These commands are similar to the previous group. Here the full scale vertical (FV) of display trace 1 (1) is autoscaled (4).

```
0
 page
 prmt LOCATE CAL SIGNAL USING CURSOR and press "CONT"
 stop
```

This is another prompt command requesting the operator to set the data cursor location. The page command forces the display to be updated before the "prompt" is put up. This guarantees that the autoscaling operation is completed and the display updated before the prompt is displayed.

```
 curv
 log
20
 tims
 prmt CALIBRATION, IN DB/1 VOLT, IN ENTRY FIELD; press "CONT"
 stop
```

This series of commands extracts the amplitude of the signal at the cursor location previously set and places it in the x scalar stack register (visible in the entry field). The logarithm of that value is calculated and subsequently multiplied by 20. This number is the amplitude of the signal at the cursor location, the calibration signal, in dB relative to one volt. As a result of the automatic stack manipulation, the result is visible in the x scalar stack register.

```
 prmt ENTER CALIBRATION SPL (IN DB) and press "CONT"
 stop
```

This "prompt" requests the calibration level in units of sound pressure level (dB relative to 20 micro pascals). This number is entered into the x scalar stack register and the previous result is automatically moved into the y scalar stack register.

```
 mins
20
 div
 pwr
```

This group of instructions extracts the calibration constant in terms of volts/20 micro pascals. This is done by subtracting (mins) x (dB/20 micro pascals) from y (db/volt). The difference is converted back to a ratio by dividing (div) the result by 20 and then taking the anti-log (pwr) of that result.

```
 CL 1
```

The result of the above operation is used as the argument in setting the input calibration constant (CL) on input channel (1).

```
"20uP"
 UN 1
 prmt RECALIBRATING DISPLAY AND CURSOR SCALING
0
 page
 FV 4 1
 FV 1 1
```

The input units (UN) on input channel 1 (1) are set to 20 micro pascals (20uP). Only the first four letters of the unit name are significant. The prompt (prmt) and the page commands are used to alert the operator without requiring intervention. The full scale display amplitude (FV) of display trace 1 (1) is autoscaled (4) and increased (1) to the next higher 10 dB value.

```
0
 page
 prmt
```

The prompt (prmt) with a null text string clears the view line of the display.

```
" READY
```

The text string (" READY) is placed in the entry field to alert the operator that the program has finished the calibration.

```
 stop
 goto DBCAL.MC
```

The stop command ends program execution. The goto command will direct the 804A to the beginning of the program in the event that the operator inadvertently presses the CONT key and prevents an "END OF FILE" error message from being displayed.

A-2. Program: DISTLIST

The following macro-program is a slightly more sophisticated programming example. This program is called DISTLIST. MC. It uses subroutines, branch on test condition commands, softkey labels, and scalar storage registers. DISTLIST is a user interactive program for determining total harmonic distortion for an input signal with a fundamental frequency range of 50 to 100 Hz. It measures harmonic power in bands about each harmonic and tests for a variety of error conditions. Here is the complete listing as it would appear in the program buffer:

```
DISTLIST:
* PROGRAM TO MEASURE TOTAL HARMONIC DISTORTION
* INPUT SIGNAL ON CHANNEL A; 50-100 HZ
* SET UP CONTROLS
"DIST.SU
 DK 2
* SELECT SINGLE DISPLAY
4
 page
* AUTOSCALE CHANNEL A INPUT SENSITIVITY
 SN   8 1
* INITIALIZE HARMONIC COUNTER,SCALAR STORAGE REGISTER, n.
2
 stor n
* INITIALIZE HARMONIC POWER ACCUMULATOR, SCALAR STORAGE REG., s.
0
 stor s
* MARKERS ON.  CLEAR MARKER LIST
 OP 5
"
 OP 7
* START AVERAGER, WAIT FOR END OF AVERAGING
 OP 1
 wait avon
 wait avon
 FH 6 1
 wait avof
* AUTOSCALE DISPLAY FULL SCALE
 FV   4 1
* IDENTIFY FUNDAMENTAL FREQUENCY
 prmt ALIGN CURSOR WITH HARMONICS.  PRESS CONT TO CONTINUE
 stop
 prmt
```

```
*SAVE FUNDAMENTAL FREQUENCY IN SCALAR STORAGE REGISTER, p
 sscl
 curh
 stor p
* MARK FUNDAMENTAL FREQUENCY
 OP 6
* SAVE 5 DELTA F IN SCALAR STORAGE REGISTER, q.
 delf
5
 tims
 stor q
*CALCULATE LOWER FREQUENCY LIMIT
 mins
 ent
 ent
*TEST FOR MIN FREQUENCY <0
 ifls ERROR
*CALCULATE UPPER FREQUENCY LIMIT
 recl q
2
 tims
 pls
* CALCULATE POWER IN FUNDAMENTAL, STORE IN SCALAR STORAGE REG., r
 rma
 stor r
* CALCULATE POWER IN EACH HARMONIC
LOOP:
* INCREMENT HARMONIC COUNTER, n
 recl n+
* CALCULATE HARMONIC FREQUENCY
 recl p
 tims
 scur
* VERIFY LESS THAN 10 MARKERS.  MARK HARMONIC
10
 ent
 recl A
 mins
 ifls ERROR2
 OP 6
 curh
```

```
MARK:
 OP 8
2000
 dely
 goto EXIT
PLOT:
 PT 1
 goto CHOICE
```

The program in the following listing is broken into functionally connected groups of commands with appropriate comments.

```
DISTLIST:
* PROGRAM TO MEASURE TOTAL HARMONIC DISTORTION
* INPUT SIGNAL ON CHANNEL A;50-100 HZ
```

The program label and initial comments.

```
* SET UP CONTROLS
"DIST.SU
 DK 2
* SELECT SINGLE DISPLAY
4
 page
```

Recalling (DK 2) a previously stored setup file, "DIST.SU", and selecting a single display (4) page.

```
* AUTOSCALE CHANNEL A INPUT SENSITIVITY
 SN   8 1
```

Autoscaling (8) the input sensitivity (SN) of input channel A (1).

```
* INITIALIZE HARMONIC COUNTER, SCALAR STORAGE REGISTER, n.
2
 stor n
* INITIALIZE HARMONIC POWER ACCUMULATOR, SCALAR STORAGE REG., s.
0
 stor s
```

Initialize scalar storage registers n and s by storing (stor) 2 and 0 in them, respectively. There are 52 scalar storage registers, a-z and A-Z. Registers A-K are reserved for storage of marker information. Register L is used to store error indications. Registers Y and Z are used for plotter scaling and offset information.

```
* MARKERS ON. CLEAR MARKER LIST
 OP 11
 OP 5
"
 OP 7
```

The data-gathering logic is released by OP 11. Markers are then turned on by OP 5 (see OPER softkeys). Previously stored markers are cleared (7) by, first, clearing the entry field using a null text string, followed by the clear command (7).

```
* START AVERAGER, WAIT FOR END OF AVERAGING
 OP 1
 wait avon
 wait avon
 FH 6 1
 wait avof
```

The operate (OP) softkeys are also used to control the averager. The averager is started (1). The wait command is used to stop program execution allowing the averager to start (avon), and to finish (avof). The second wait avon command followed by the command to view the full (6) horizontal scale (FH) on display trace 1 (1) is required to make sure that not only has the averager started, but all tasks have had a chance to run so that information needed to determine the horizontal limits are set up. The second wait could be replaced with "0 page" which forces the data processing and display formatting tasks to run.

```
* AUTOSCALE DISPLAY FULL SCALE
 FV   4 1
```

These commands autoscale (4) the vertical fullscale (FV) amplitude on display trace 1 (1)

```
* IDENTIFY FUNDAMENTAL FREQUENCY
 prmt ALIGN CURSOR WITH HARMONICS.  PRESS CONT TO CONTINUE
 stop
 prmt
```

A prompt (prmt) requests the operator to align the data cursor with the harmonics on the display. The stop command halts program execution until the operator acknowledges the message by pressing the CONT key. The prompt command with the null text string clears the view line when program execution resumes.

```
*SAVE FUNDAMENTAL FREQUENCY IN SCALAR STORAGE REGISTER, p
 sscl
 curh
 stor p
```

This group of commands selects scalar math mode (sscl) and enters the horizontal location of the data cursor (curh) into the x scalar stack register. The cursor horizontal position is also stored (stor) in scalar storage register p.

```
* MARK FUNDAMENTAL FREQUENCY
 OP 6
```

Set (6) a marker (OP) at the cursor location.

```
* SAVE 5 DELTA F IN SCALAR STORAGE REGISTER, q.
 delf
5
 tims
 stor q
```

Enter the frequency resolution delta f (delf) into the x scalar stack register. Multiply (tims) that value by 5 and store (stor) the result in scalar storage register q.

```
*CALCULATE LOWER FREQUENCY LIMIT
 mins
 ent
 ent
```

Subtract (mins) the value 5 delta f from the fundamental frequency previously entered into the scalar math stack. Enter (ent) the difference twice. This moves it into the y and z scalar stack registers as well as the x register.

```
*TEST FOR MIN FREQUENCY <0
 ifls ERROR
```

If the contents of the x scalar stack register are less than zero (ifls), then transfer program execution (branch) to a program labeled ERROR. This test guarantees that the fundamental is not within 5 delta f of 0 Hz (D.C.).

```
*CALCULATE UPPER FREQUENCY LIMIT
 recl q
2
 tims
 pls
```

These commands recall (recl) the contents of scalar storage register q (5 delta f). This number is entered into the x stack register and multiplied (tims) by 2. The product is added (pls) to the contents of the y scalar stack register which is fundamental frequency minus 5 delta f. The result is the fundamental frequency plus 5 delta f.

```
* CALCULATE POWER IN FUNDAMENTAL,STORE IN SCALAR STORAGE REG., r
 rma
 stor r
```

The x scalar stack register currently holds 5 delta f plus the fundamental frequency. The y stack register contains the fundamental frequency minus 5 delta f. These limits define a frequency band. The contents of the x and y stack registers are arguments for the rma command. This command returns the sum of power in the frequency band which brackets the fundamental frequency. The sum is entered into the x scalar stack register by the rma command. It is also stored (stor) in scalar storage register r.

```
* CALCULATE POWER IN EACH HARMONIC
LOOP:
```

Since there will be more than one harmonic, the program incorporates a loop. The label LOOP is an entry point to this loop.

```
* INCREMENT HARMONIC COUNTER, n
 recl n+
```

The contents of scalar storage register n, used as the harmonic counter, is recalled and the storage register contents incremented (recl n+). The x scalar stack register now contains the harmonic number.

```
* CALCULATE HARMONIC FREQUENCY
 recl p
 tims
 scur
```

The fundamental frequency stored in scalar storage register p is recalled (recl) to the x stack register. The stack has auto-entered the previous data, moving the harmonic number to the y stack register. The contents of the x and y stack registers are multiplied (tims). The result is used to set the cursor (scur) to the harmonic frequency.

```
* VERIFY LESS THAN 10 MARKERS. MARK HARMONIC
10
 ent
 recl A
 mins
 ifls ERROR2
 OP 6
 curh
```

This group of commands recalls (recl) the total number of markers currently set, which are stored in storage register A. It tests if more than ten markers are set. If all the markers are used, it branches to program ERROR2. If markers are available, it sets (6) a marker (OP) at the current harmonic frequency. The x scalar stack register is then reloaded with the harmonic frequency by reading the current cursor horizontal location (curh).

```
* CALCULATE LOWER LIMIT OF FREQUENCY BAND
 recl q
 mins
 ent
 ent
 recl q
2
 tims
 pls
 ent
 ent
```

This set of commands recalculates the frequency band about the current harmonic frequency.

```
*TEST FOR MAX FREQUENCY>SPAN IF SO EXIT
 fspa
 mins
 ifgt EXIT
 roll
```

The frequency span is entered into the x stack register (fspa) and subtracted (mins) from the upper band limit. The difference, if greater than zero (ifgt), causes the program to branch to EXIT. The stack is rolled (roll) to return the band limits to the x and y stack registers.

```
* CALCULATE POWER IN HARMONIC n
 rma
 recl s
 pls
 stor s
```

The power in a band defined by the frequency limits stored in the x and y stack registers is calculated (rma) and stored in the x stack register. The current contents of scalar storage register s, being used to hold total harmonic power, are recalled (recl) and added (pls) to the power stored in the x stack register. The sum is stored (stor) in the s storage register.

```
* RETURN FOR NEXT HARMONIC
 goto LOOP
```

The unconditional branch command (goto) forces the program back to the entry point, LOOP. The program will continue to execute between the label LOOP and goto LOOP until the upper frequency limit of the band exceeds the frequency span, a test which was performed earlier.

```
* CALCULATE HARMONIC DISTORTION
EXIT:
 recl s
 recl r
 div
 sqrt
100
 tims
```

This program segment, with the entry point label EXIT, calculates the total harmonic distortion based on data stored in the scalar storage registers. Total harmonic power is recalled (recl) from storage register s. Power of the fundamental is recalled from storage register r. After the recall operations, this data resides in the y and x stack registers, respectively. The harmonic power is divided (div) by the fundamental power. The square root (sqrt) of the result is calculated and returned to the x stack register. The result of this operation is multiplied (tims) by 100 and returned to the x stack register. This value is the total harmonic distortion in percent.

```
PROMPT:
 prmt HARMONIC DISTORTION(PERCENT)IN ENTRY FIELD
```

This prompt (prmt) identifies the data in the entry field.

```
CHOICE:
 key1  PLOT   REPEAT  LIST
 soft PLOT DISTLIST MARK
 goto PROMPT
```

This command group is labeled (key1) and displays three softkeys, PLOT, REPEAT, and LIST. The program will branch based on the operator selection of the softkeys (soft). The corresponding destination entry points will be PLOT, DISTLIST, and MARK. The goto command returns the program to the entry point PROMPT if any key other than the three labeled softkeys is pressed. This will repeat the softkey selection operation until one of the correct keys is pressed. Although not done here, it is good programming practice to include a stop or abort choice at all such user decision points. This is true for any computer.

```
ERROR:
 prmt FUNDAMENTAL FREQUENCY TOO LOW
 stop
 goto DISTLIST
```

This is an error-handling routine. If the lower band limit which was tested earlier is zero or negative, then the program would have branched to this entry point. A prompt (prmt) would alert the operator and terminate program execution by use of the stop command. The unconditional branch (goto) re-starts the program at the beginning (DISTLIST) in the event that the CONT key is pressed. This prevents the following program segments from being executed accidentally.

```
ERROR2:
 recl A
 prmt TOO MANY MARKERS
 stop
 goto DISTLIST
```

This is another error-handling routine. It recalls (recl) the number of markers set (A) and puts that number into the x stack register (entry field). It notifies the operator with a prompt (prmt) and stops program execution. The unconditional branch (goto) again prevents the program from accidentally passing this point.

```
MARK:
 OP 8
0
 page
 goto EXIT
```

This section of the program will display the marker list (OP 8). The command "0 page" forces the display to be updated. The branch command (goto) directs the program to the EXIT routine. This will recalculate distortion and display the associated prompt.

```
PLOT:
 PT 1
 goto CHOICE
```

This final program segment starts (1) the plotter (PT) to produce a plot of the display. The unconditional branch (goto) directs the program back to the entry point CHOICE.

This concludes the Programming Example Section. Part 2, called the 804A Programmer's Handbook, provides a one page review of SAMPL syntax followed by lists of all SAMPL basic and extended commands in alphabetical order and by functional group. All possible parameters for each command are listed.

ENTRY AND NOTATION

4-1. 804A Program Entry

All numeric entries without a leading space are automatically entered into the entry field.

In SAMPL™ (Signal Analysis Macro Programming Language), alphanumeric entries which are not preceded by a space and are followed by a colon (:) are program labels.

All alphanumeric entries preceded by a space are interpreted as command mnemonics.

Other significant first characters are:

* Non-executable comment line follows.

" String delimiter. Used to indicate a string which is neither command nor label in a macro. Remotely entered alphanumeric strings without quotes are assumed to be command mnemonics. Strings are terminated by another quote or by end of line.

4-2. Notation of this Handbook

Several letters will be used to represent the arguments of those commands which require arguments. These letters are not meant to be typed in as part of the command. They should be replaced by an argument of the appropriate type.

N - Number in Entry Field.

C - Softkey choice. Choices for each command will be spelled out in this handbook. C ranges between one and sixteen.

S - Select input channel, memory, or trace affected. Always 1, 2, 3, or 4.

T - Text, such as key labels or disk file name, etc.

CHAPTER 5

MNEMONICS

5-1. Basic Operations

All basic operations (operations which can be performed from the primary keyboard without entering alpha or math shift) are represented by two letter mnemonics. Both letters are capitalized. Remember that ALL MNEMONICS MUST BE PRECEDED BY A SPACE in a program.

Basic commands will be presented in alphabetical order in Section 1 of 5-1 and grouped by function in Section 2 of 5-1 (page 53).

1. Alphabetical Order

Entry/ Mnemonic	Description	Notes
N AB	A-B,C,D Delay	N is time delay in seconds.
N AC	Annotation Plot Color	N is 1 to 8, pen color
AH C	Auto Horizontal Scaling	C: 1 = Off, 2 = On
AM C	Averager Mode	Choices listed below 1. SU 5. EX TR 2. SU TR 6. PK 3. SU PRV 7. PK TR 4. EX 8. PK PRV
N AN	Number of Averages	N from 1 to 10000
AO C	Annotation State	C: 1 = Off, 2 = On
AT C	Averaging Type (domain)	C: 1 = Time, 2 = Freq

Entry/ Mnemonic	Description	Notes
N CC	Cursor Plot Color	N is 1 to 8, Pen Color
CH C	Number of Channels	C: 1 = 2CH, 2 = 4CH
N CL S	Input Calibration	S : 1=A, 2=B, 3=C, 4=D
CT C	Cursor type	Choices listed below (choices for single display) 1. SNGL 2. SIDEBD 3. HAR (Choices for dual display) 1. SNGL 5. 2nd 2. SIDEBD 6. Both 3. HAR 7. Copy up 4. 1st 8. Copy dn (Choices for quad display) 1. SNGL 5. Dsp 2 2. SIDEBD 6. Dsp 3 3. HAR 7. Dsp 4 4. Dsp 1
DA C	Data State (display)	C: 1 = Off, 2 = On
N DC S	Data Plot Color	N is 1 to 8, Pen Color S is Trace #, 1 to 4

Note: For HP-GL plotters, N may be a three digit number to specify pattern length, line type and color.
1st digit - length of pattern as a percent of upper right minus lower left.
2nd digit - Pattern.
3rd digit - pen color, 1 to 8 as above.

Entry/ Mnemonic	Description	Notes
DD C T	Dual Display Pair Selection	(Choices listed below.) 1. 1&2 5. 2&4 2. 1&3 6. 3&4 3. 1&4 4. 2&3
DK C	Disk functions	T is a File Name or Template. 1. STORE 5. REPLACE 2. RECALL 3. LIST 4. DELETE
DN C	Single display select	C: Trace #, 1 to 4
DP C S	Copy/swap trace	S: Trace #, 1 to 4. (Choices listed below.) 1. copy 1 5. swap 1 2. copy 2 6. swap 2 3. copy 3 7. swap 3 4. copy 4 8. swap 4
DS C	Disk store selection	Choices as below (If "MENU" selected DS 2) 1. VIEW 7. AUX 2. MENU 8. ALL 3. MEM 9. FULL 4. TRACE 10. PROG 5. MEAS 11. TEXT 6. DSP 12. AVER (If "MEM" selected DS 3) 1. VIEW 9. FULL 2. MENU 10. PROG 3. MEM 11. TEXT 4. TRACE 12. AVER 5. X 13. 6. Y 14. 7. Z 15. V1-4 8. T 16. XYZT

Entry/ Mnemonic	Description	Notes
		(If "TRACE" selected DS 4) 1. VIEW 9. FULL 2. MENU 10. PROG 3. MEM 11. TEXT 4. TRACE 12. AVER 5. 1 13. 6. 2 14. 7. 3 15. 8. 4 16. ALL Else the right four columns are empty.
DU C S	Memory Update State	C: 1= Live, 2= Freeze S: 1=X, 2=Y, 3=Z, 4=T
DW C	Data Weighting	(Choices listed below.) 1. FLAT 5. IMP BCD 2. HANN 6. IMP ALL 3. FLTOP 4. IMP A
FH C S	Horizontal Full Scale	S: Trace #, 1 to 4 (Choices listed below.) 1. INCR 5. MAG BD 2. DECR MK 1-2 3. SET TO CURSOR 6. FULL VIEW
FL C	Frequency location	C: 1=start, 2=center
FM C S	Format Vertical, Horizontal	S: trace #, 1 to 4 (Choices listed below.) 1. LN,LN 9. LN,LG 2. LG,LN 10. LG,LG 3. R,LN 11. R,LG 4. I,LN 12. I,LG 5. PH,LN 13. PH,LG

Entry/ Mnemonic	Description	Notes
FU C S	Display Function	S: 1=X, 2=Y, 3=Z, 4=T (Choices listed below.) 1. ITIME 7. C 2. ATIME 8. D 3. ISPEC 9. PWR 4. RMS 10. CROSS 5. A 11. XFER 6. B 12. COH
FV C S	Vertical Full Scale	S: Trace #, 1 to 4 (Choices listed below.) 1. INCR 5. ZERO 2. DECR CENTER 3. SET TO CURSOR 6. FULL VIEW 4. AUTO SCALE
N GC	Grid Color	N: Pen number, 1 to 8 (See Note for "DC", above)
GR C	Grid State	C: 1=ON, 2=OFF, 3=Border
N HF S	Horizontal Full Scale	N: Full scale in Hz, Orders seconds, etc. S: Trace #, 1 to 4
N HM S	Horizontal Minimum	N: Minimum horizontal position S: Trace #, 1 to 4 N
HR S	Horizontal Reference	N: Horizontal reference S: Trace #, 1 to 4
HU C S	Horizontal units	S: Trace #, 1 to 4 Choices: Time: 1. SEC 9. SEC REL 2. MIN 10. MIN REL 3. CYC 11. CYC REL Frequency: 1. Hz 9. Hz REL 2. CPM 10. CPM REL 3. ORD 11. ORD REL
N IO	IEEE-488 remote control I/O address	N: 0 to 30

Entry/ Mnemonic	Description	Notes
IT C S	Input Coupling	C: 1 = AC, 2 = DC S: Channel 1=A to 4=D
N MC	Marker Color	N: Pen color, 1 to 8 Note: N may be a three digit number; see "DC".
MD C	Coverage Mode	C: 1 = Baseband 2 = Zoomband
MH C S	Minimum Horizontal (See also "HM", for direct entry of minimum horizontal.)	S: Trace #, 1 to 4 (Choices listed below.) 1. INCR 2. DECR 3. SET TO CURSOR 4. 5. MAG BD MK 1-2 6. FULL VIEW
MM C	Set Up Memory (Set up memory is made up of all menus and scalar registers.)	C: 1=save, 2=swap
MV C S	Minimum Vertical	S: Trace #, 1 to 4 Choices; linear, real, imag, phase: 1. INCR 2. DECR 3. SET TO CURSOR 4. AUTO SCALE 5. ZERO CENTER 6. FULL VIEW

Entry/ Mnemonic	Description	Notes
MV C S (cont'd.)		Choices, log magnitude: 1. 10dB 9. 90dB 2. 20dB 10. 100dB 3. 30dB 11. 110dB 4. 40dB 12. 120dB 5. 50dB 13. 130dB 6. 60dB 14. 140dB 7. 70dB 15. 150dB 8. 80dB 16. 160dB
N NL	Noise level	N: Volts Peak-Peak. 0<N<20
NT C	Noise type	C: 1=random, 2=repeat, 3=sine
OP C	Operate	(Choices listed below.) ----AVERAGE---- 1. START 9. STOP 2. CONTIN ----CAPTURE---- 3. HOLD 11. ARM 4. REL 12. TRIG ----MARKERS---- 5. ON 13. OFF 6. SET 7. CLR 8. LIST
N OS	Order scaling	N: Hz/order scale factor

Entry/ Mnemonic	Description	Notes
N PA	Plotter Address on 488 N: 0 to 30	
PN C S	PSD Normalization	C: 1 = off, 2 = on S: Memory 1=X,2=Y,3=Z,4=T.
PT C	Plot State	(Choices listed below.) 1. START 5. CLEAR 2. STOP SCALE 3. SET X 4. X SET
PV	Plotter Page Advance	N: 0 to 4, number of pages to advance. Should be 0 if plotter lacks page advance capability.
PY C	Plotter Type	C: 1 = HPGL, 2 = Plot 50
RD C	Redundancy	(Choices listed below.) 1. MAX 3. 50% 2. 75% 4. 0%
RH C S	Horizontal Reference	S: Trace #, 1 to 4 Choices: 1. SET=1 5. SET TO 2. SET=2 CURSOR 3. SET=3 4. SET=4
RI C	Remote Control Interface	C: 1 = RS232, 2 = IEEE-488 3 = none

Entry/ Mnemonic	Description	Notes
RV C S	Vertical Reference	(Choices listed below.) 1. SET=1 2. SET=2 3. SET=3 4. SET=4 5. SET TO CURSOR
SD C S	Data source	S: Trace #, 1 to 4 C: Memory, 1=X to 4=T
SM C	Sampling source	C: 1=internal, 2=external
SN C S	Input sensitivity	S: Channel, 1=A to 4=D (Choices listed below.) 1. 2 mV 2. 5 mV 3. 10 mV 4. 20 mV 5. 50 mV 6. 100mV 7. 8. AUTO SCALE 9. 200mV 10. 500mV 11. 1 V 12. 2 V 13. 5 V 14. 10 V
SP C	Coverage span	(Choices listed below.) 1. 1 HZ 2. 2 HZ 3. 5 HZ 4. 10 HZ 5. 20 HZ 6. 50 HZ 7. 100 HZ 8. 200 HZ 9. 500 HZ 10. 1 KHZ 11. 2 KHZ 12. 5 KHZ 13. 10 KHZ 14. 20 KHZ 15. 40 KHZ

Note: 40 KHz (SP 15) not available in Zoom band.

Entry/ Mnemonic	Description	Notes
N TA	Trigger to A delay	N: Delay in memory windows -1 to +63
N TA	Trigger Level	N: Trigger level as % of for external trigger is 5 V).
TP C	Trigger slope	C: 1 = +, 2 = -
TS C	Trigger source	(Choices listed below.) 1. A 4. D 2. B 5. EXT 3. C
TX C	Text, Titles	(Choices listed below.) 1. FIRST 5. TEXT 2. LAST ---AREA--- ---PAGE--- 6. PROG 3. PREV 7. AUX 4. NEXT Note: EDIT not available from a program.
T UF S	Function Label Note: Works only on frozen memories.	T: up to 12 characters S: Memory, 1=X,2=Y,3=Z,4=T.
T UN S	Input units	T: Units, up to 4 characters S: Channel, 1=A,2=B,3=C,4=D.
N VF S	Vertical full scale See also "FV"	N: full scale value S: Trace #, 1 to 4
N VM S	Vertical minimum See also "MV"	N: minimum value S: Trace #, 1 to 4
N VR S	Vertical reference	N: reference value S: Trace #, 1 to 4

Entry/ Mnemonic	Description	Notes
VS C	Vector Storage from X	(Choices listed below.) 1. SAVE V1 5. SWAP V1 2. SAVE V2 6. SWAP V2 3. SAVE V3 7. SWAP V3 4. SAVE V4 8. SWAP V4 Note that "recall" is implemented as SWAP n, SAVE n.
VU C S	Vertical units	S: Trace #, 1 to 4 (Choices listed below.) 1. LN 9. LN REL 2. LG 10. LG REL

2. Order by Functional Group

This section presents the same commands as Part 2 of Section 5-1. grouped by function. Some commands appear in two or three functional groups.

Commands within each group are listed in alphabetical order.

A. System Configuration Commands

This group contains commands which are generally used before any data processing, such as selecting two or four channel operations, associating functions with memories and memories with display traces, and selecting AC or DC coupling.

Enty/ Mnemonic	Description	Notes
CH C	Number of channels	C: 1 = 2CH, 2 = 4CH
DP C S	Copy/swap trace	S: Trace #, 1 to 4. (Choices listed below.) 1. copy 1 5. swap 1 2. copy 2 6. swap 2 3. copy 3 7. swap 3 4. copy 4 8. swap 4

Entry/ Mnemonic	Description	Notes
DU C S	Memory update state	C: 1= Live, 2 = Freeze S: 1=X, 2=Y, 3=Z, 4=T
FU C S	Display function	S: 1=X, 2=Y, 3=Z, 4=T (Choices listed below.) 1. ITIME 7. C 2. ATIME 8. D 3. ISPEC 9. PWR 4. RMS 10. CROSS 5. A 11. XFER 6. B 12. COH
IT C S	Input coupling	C: 1 = AC, 2 = DC S: Channel 1=A to 4=D
SD C S	Data source	S: Trace #, 1 to 4 C: Memory, 1=X to 4=T

B. Averager Control Commands

Commands in this group affect the operation of the averager.

Entry/ Mnemonic	Description	Notes
AM C	Averager Mode	(Choices listed below.) 1. SU 5. EX TR 2. SU TR 6. PK 3. SU PRV 7. PK TR 4. EX 8. PK PRV
N AN	Number of Averages	N from 1 to 10000
AT C	Averaging Type(domain)	C: 1 = Time, 2 = Freq

Entry/ Mnemonic	Description	Notes
OP C	Operate	(Choices listed below.) ----AVERAGE---- 1. START 9. STOP 2. CONTIN ----CAPTURE---- 3. HOLD 11. REL 4. ARM 12. TRIG 5. ON 13. OFF 6. SET ----MARKERS---- 7. CLR 8. LIST
RD C	Redundancy	(Choices listed below.) 1. MAX 3. 50% 2. 75% 4. 0%

C. Display Control Commands

Commands in this group affect the format of the display. Please also see the extended operation "page".

Entry/ Mnemonic	Description	Notes
AH C	Auto Horizontal Scaling	C: 1 = Off, 2 = On
DD C	Dual Display Pair Selection.	(Choices listed below.) 1. 1&2 5. 2&4 2. 1&3 6. 3&4 3. 1&4 4. 2&3
DN C	Single Display Select	C: Trace #, 1 to 4

Entry/ Mnemonic	Description	Notes
FH C S	Horizontal Full Scale	S: Trace #, 1 to 4 (Choices listed below.) 1. INCR 2. DECR 3. SET TO CURSOR 5. MAG BD MK 1-2 6. FULL VIEW
FM C S	Format Vertical,	S: trace #, 1 to 4 (Choices listed below.) 1. LN,LN 2. LG,LN 3. R,LN 4. I,LN 5. PH,LN 9. LN,LG 10. LG,LG 11. R,LG 12. I,LG 13. PH,LG
FU C S	Display function	S: 1=X, 2=Y, 3=Z, 4=T (Choices listed below.) 1. ITIME 2. ATIME 3. IAPEC 4. RMS 5. A 6. B 7. C 8. D 9. PWR 10. CROSS 11. XFER 12. COH
FV C S	Vertical Full Scale	S: Trace #, 1 to 4 (Choices listed below.) 1. INCR 2. DECR 3. SET TO CURSOR 4. AUTO SCALE 5. ZERO CENTER 6. FULL VIEW
N HF S	Horizontal Full Scale	N: Full scale in Hz, Orders seconds, etc. S: Trace #, 1 to 4
N HM S	Horizontal minimum	N: Minimum horizontal position S: Trace #, 1 to 4

Entry/ Mnemonic	Description	Notes
MH C S	Minimum horizontal	S: Trace #, 1 to 4 (Choices listed below.) 1. INCR 5. MAG BD 2. DECR MK 1-2 3. SET TO 6. FULL CURSOR VIEW 4. (See also "HM", for direct entry of minimum horizontal.)
MV C S	Minimum Vertical	S: Trace #, 1 to 4 Choices: linear, real, imag, phase: 1. INCR 5. ZERO 2. DECR CENTER 3. SET TO 6. FULL CURSOR VIEW 4. AUTO SCALE Choices: log magnitude: 1. 10 dB 9. 90 dB 2. 20 dB 10. 100dB 3. 30 dB 11. 110dB 4. 40 dB 12. 120dB 5. 50 dB 13. 130dB 6. 60 dB 14. 140dB 7. 70 dB 15. 150dB 8. 80 dB 16. 160dB
PN C S	PSD normalization	C: 1 = off, 2 = on S: Memory 1=X,2=Y,3=Z,4=T.
T UF S	Function label Note: Works only on frozen memories.	T: up to 12 characters S: Memory, 1=X,2=Y,3=Z,4=T.
N VF S	Vertical full scale	N: full scale value
N VM S	Vertical minimum	N: minimum value S: trace #, 1 to 4

D. Calibration and Reference Commands

Commands in this group allow the definition of input calibration constants specifying volts per unit (such as 0.100 volts per g), specification of a units label to be displayed with amplitude readings, selection of horizontal units, definition of references, etc.

Entry/ Mnemonic	Description	Notes
N CL S	Input calibration	S : 1=A, 2=B, 3=C, 4=D
N HR S	Horizontal reference	N: Horizontal reference S: Trace #, 1 to 4
HU C S	Horizontal units	S: Trace #, 1 to 4 (Choices listed below.) Time: 1. SEC 9. SEC REL 2. MIN 10. MIN REL 3. CYC 11. CYC REL Frequency: 1. HZ 9. HZ REL 2. CPM 10. CPM REL 3. ORD 11. ORD REL
RH C S	Horizontal reference	S: Trace #, 1 to 4 (Choices listed below.) 1. SET=1 4. SET=4 2. SET=2 5. SET TO 3. SET=3 ORDER
RV C S	Vertical reference	S: Trace #, 1 to 4 (Choices listed below.) 1. SET=1 4. SET=4 2. SET=2 5. SET TO 3. SET=3 CURSOR
T UN S	Input units	T: Units, up to 4 characters S: Channel, 1=A to 4=D
N VR S	Vertical reference	N: reference value S: Trace #, 1 to 4

Entry/ Mnemonic	Description	Notes
VU C S	Vertical units	S: Trace #, 1 to 4 (Choices listed below.) 1. LN 9. LN REL 2. LG 10. LG REL

E. Cursor and Marker Control Commands

Commands in this section turn on and off the cursor and markers. To move the cursor to a specific position see the extended commands "scur", "scel", "dcrt", and "dclf". The extended commands "dcon" and "dcof" can also be used to turn the data cursor on and off.

Entry/ Mnemonic	Description	Notes
CT C	Cursor type	Choices listed below. (Choices for single display:) 1. SNGL 2. SIDEBD 3. HAR (Choices for dual display:) 1. SNGL 5. 2nd 2. SIDEBD 6. BOTH 3. HAR 7. COPY UP 4. 1st 8. COPY DN (Choices for quad display:) 1. SNGL 5. DSP 2 2. SIDEBD 6. DSP 3 3. HAR 7. DSP 4 4. DSP 1
OP C	Operate	(Choices listed below.) ----AVERAGE---- 1. START 9. STOP 2. CONTIN ----CAPTURE---- 3. HOLD 11. REL 4. ARM 12. TRIG ----MARKERS---- 5. ON 13. OFF 6. SET 7. CLR 8. LIST

F. Transient Capture and Delay Commands

Commands in this section control the conditions for and the effect of triggering.

Entry/ Mnemonic	Description	Notes
N AB	A-B,C,D Delay	N is time delay in seconds.
OP C	Operate	(Choices listed below.) ----AVERAGE---- 1. START 9. STOP 2. CONTIN ----CAPTURE---- 3. HOLD 11. REL 4. ARM 12. TRIG ----MARKERS---- 5. ON 13. OFF 6. SET 7. CLR 8. LIST
N TA	Trigger to A delay	N: Delay in memory windows -1 to +63
N TL	Trigger level	N: Trigger level as % of full scale (full scale for external trigger is 5 V).
TP C	Trigger slope	C: 1 = +, 2 = -
TS C	Trigger source	(Choices listed below.) 1. A 4. D 2. B 5. EXT 3. C

G. Input Sensitivity and Coupling

Entry/ Mnemonic	Description	Notes
IT C S	Input coupling	C: 1 = AC, 2 = DC S: Channel 1=A,2=B,3=C,4=D.
SN C S	Input sensitivity	S: Channel, 1=A to 4=D (Choices listed below.) 1. 2 MV 8. AUTO SCALE 2. 5 MV 3. 10 MV 9. 200MV 4. 20 MV 10. 500MV 5. 50 MV 11. 1 V 6. 100MV 12. 2 V 7. 13. 5 V 14. 10 V

H. Weighting

Commands in this section affect the weighting window applied to the time data. Flat weighting is used for transient capture and for repeating noise. Hanning weighting is generally used for continuous periodic signals. Impulse weighting is used almost exclusively for impact testing objects which ring for longer that one window. Flat top weighting provides worse skirt suppression than Hanning weighting, but has practically no "picket fence" effect. Flat top weighting should, therefore, be used when precise readings of amplitude are required for continuous signals. Please see the 804A manual for details.

Entry/ Mnemonic	Description	Notes
DW C	Data Weighting	(Choices listed below.) 1. FLAT 5. IMP BCD 2. HANN 6. IMP ALL 3. FLTOP 4. IMP A

I. Frequency and Order Controls

External sampling is used for order tracking. Commands in this group allow internal or external sampling, allow the display to read orders, and specify baseband and zoom coverage selections.

Entry/ Mnemonic	Description	Notes
FL C	Frequency location	C: 1 = start, 2 = center
N FQ	Zoom origin/center	N = freq. in Hertz
MD C	Coverage mode	C: 1 = Baseband 2 = Zoomband
N OS	Order scaling	N: Hz/order scale factor
SM C	Sampling source	C: 1=internal, 2=external
SP C	Coverage span	(Choices listed below.) 1. 1 HZ 8. 200HZ 2. 2 HZ 9. 500HZ 3. 5 HZ 10. 1 KHZ 4. 10 HZ 11. 2 KHZ 5. 20 HZ 12. 5 KHZ 6. 50 HZ 13. 10 KHZ 7. 100HZ 14. 20 KHZ 15. 40 KHZ Note: 40 KHz (SP 15) not available in Zoom band.

J. Disk and Other Storage Controls

Commands in this group control what is to be stored on disk when disk store operation is performed, various disk functions, and storage in memory such as the four vector storage buffers and the setup memory. See also the extended operations "stor" and "recl" for storage of scalars.

Entry/ Mnemonic	Description	Notes
T DK C	Disk functions	T is a file name or template. 1. STORE 4. DELETE 2. RECALL 5. REPLACE 3. LIST
DS C	Disk store selection	(Choices listed below.) (If "MENU" selected DS 2). 1. VIEW 7. AUX 2. MENU 8. ALL 3. MEM 9. FULL 4. TRACE 10. PROG 5. MEAS 11. TEXT 6. DSP 12. AVER (If "MEM" selected (DS 3) 1. VIEW 9. FULL 2. MENU 10. PROG 3. MEM 11. TEXT 4. TRACE 12. AVER 5. X 15. V1-4 6. Y 16. ALL 7. Z 8. 4 (If "TRACE" selected (DS 4) 1. VIEW 7. 3 2. MENU 8. 4 3. MEM 9. FULL 4. TRACE 10. PROG 5. 1 11. TEXT 6. 2 12. AVER 16. ALL

*Else the right four columns are empty.

Entry/ Mnemonic	Description	Notes
MM C	Set up memory (Set up memory is made up of all menus and scalar registers.)	C: 1=save, 2=swap
VS C	Vector Storeage from X	(Choices listed below.) 1. SAVE V1 5. SWAP V1 2. SAVE V2 6. SWAP V2 3. SAVE V3 7. SWAP V3 4. SAVE V4 8. SWAP V4 Note that "recall" is implemented as SWAP n, SAVE n.

K. Plotter Control Commands

Commands in this group control plot colors (and line textures, on HP-GL plotters), enable plot positioning from a SAMPL program and start and stop the plotter. Some of the commands in this section affect the display in ways that are used when plotting, such as turning off the 804A's annotation. It is also possible to overlap data and user generated text on one display (and, therefore, one plot). (See the the extended command "data".)

Entry/ Mnemonic	Description	Notes
N AC	Annotation Plot Color	N is 1 to 8, Pen Color
AO C	Annotation State	C: 1 = Off, 2 = On
N CC	Cursor Plot Color	N is 1 to 8, Pen Color (See note for DC below.)
DA C	Data State (display)	C: 1 = Off, 2 = On
N DC S	Data Plot Color	N is 1 to 8, Pen Color S is Trace #, 1 to 4 (See note on next page)

Entry/ Mnemonic	Description	Notes
		Note: For HP-GL plotters, N may be a three digit number to specify pattern length, line type and color. 1st digit - length of pattern as a percent of upper right minus lower left. 2nd digit - Pattern. 3rd digit - pen color, 1 to 8 as above.
N GC	Grid color	N: Same as for "DC", above
GR C	Grid state	C: 1=ON, 2=OFF, 3=Border
N MC	Marker color	N: Pen color, 1 to 8 Note: N may be a three digit number; see "DC".
N PA	Plotter address on 488 N: 0 to 30	
PT C	Plot State	(Choices listed below.) 1. START 5. CLEAR 2. STOP SCALE 3. SET X 4. X SET
N PV	Plotter page advance	N: 0 to 4, number of pages to advance. Should be 0 if plotter lacks page advance capability.
PY C	Plotter type	C: 1 = HPGL, 2 = Plot 50

L. Noise Source Control

Entry/ Mnemonic	Description	Notes
N NL	Noise level	N: Volts Peak-Peak. 0<N<20
NT C	Noise type	C: 1=random, 2=repeat, 3=sine

M. Text and Program Entry Control

Entry/ Mnemonic	Description	Notes
TX C	Text, Titles	(Choices listed below.) ---PAGE--- 1. FIRST 2. LAST 3. PREV 4. NEXT ---AREA--- 5. TEXT 6. PROG 7. AUX Note: EDIT not available in a program.

CHAPTER 6

EXTENDED MNEMONICS

6-1. Extended Operation Mnemonics

All extended operations (operations which cannot be performed from the primary keyboard without entering alpha or math shift) are represented by three or four letter mnemonics. All letters of extended mnemonics are lower case. Remember that ALL THESE MNEMONICS MUST BE PRECEDED BY A SPACE in a program.

1. Alphabetic listing

Entry/ Mnemonic	Description	Notes
box	Generate a box.	Amplitude is 1 between limits of Real(x) and Real(y) on X axis, else amplitude = 0.
bus T	Send IEEE-488 bus commands. Addressed commands are sent to the REMOTE CONTROL device declared on the AUX menu page, unless otherwise noted. T can be: rtl - Returns the 804a to local mode ifc - pulses interface clear rem - asserts remote enable loc - opposite of rem llo - sends local lockout (addressed command) gtl - sends go to local (addressed command) sdc - sends selected device clear (addressed command) dcl - sends device clear get - sends group execute trigger (addressed command)	
clbl T	Cursor label. "clbl" with no text recalls function code and previous T (cursor label and comment) to the prompt line. Affects memory X only; Use roll to affect the other memories.	T: first nine characters used as amplitude units for cursor readout; up to 43 additional characters are remembered as comment.

Entry/ Mnemonic	Description	Notes
clr T	Erase named structure.	Prog, text, auxa, aver x: X if svec, x if sscl, both if mixed mode. stk: Vector stack if svec, scalar stack if sscl, both if mixed mode.
cnst	Generate a vector from a constant.	x -> X
conj	Complex conjugate.	Scalar mode: x* -> x Mixed/vec: X* -> X
cont	Continue	Can not be programmed.
copy T T	Copy one file to another. Both files must exist.	T: filename
cos	Cosine x,X in radians.	Scalar mode: cos(x) -> x Mixed/vec: cos(X) -> X
curh	Cursor horizontal position goes into x. See also "gcel".	
curv	Amplitude at cursor goes into x. See also "gety".	
data	Turn on display of data traces and graticule last formatted. Used to display data over text page.	
datm	Display date and time, date and time goes into x.	Real x = month.day Imag x = hour
dclf	Move data cursor left.	
dcof	Data cursor off.	
dcon	Data cursor on.	

Entry/ Mnemonic	Description	Note
dcrt	Move data cursor right.	
delf	Frequency resolution (delta frequency) goes into x.	
delt	Time resolution goes into x.	
dely	Delay	Real(x) is delay in millisec.
div	Divide	Scalar mode: x= y/x Mixed mode: X= X/x Vector mode: X= Y/X
djw	Divide by jω pseudo integration.	Scalar mode: x= x/jω where w = cursor freq. Mixed, vec: X= X/jω
ent	Enter	See Chapter 7.
exc	Exchange X and Y.	See Chapter 7.
exp	Exponentiation	Scalar: $x = e^x$ Vector or Mixed: $X = e^X$
fft	Fast Fourier transform X = F(X), time domain data only.	
fspa	Frequency span to x.	
gcel	Get the cell # corresponding to the current cursor position. Result goes into Re(x).	
gety	Get amplitude of cell # in x. Amplitude goes into y.	
gosb T	Go to subroutine.	T: label in program
goto T	Unconditional jump.	T: label in program
help	List mnemonics.	"help" info --> aux text
help <command>	Display information on one command.	

Entry/ Mnemonic	Description	Note
herr	Errors are "hard." Hard errors stop SAMPL program execution. This is the default established by RUN. See "serr".	
ifeq T	If the real part of x = 0, then branch to statement label T.	
ifft	Inverse Fourier transform on X.	
ifge T	If Real(x) is greater than or equal to zero then branch to statement label T.	
ifgt T	If Real(x) is greater than zero, then branch then branch to statement label T.	
ifle T	If Real(x) less than or equal to zero then branch, then branch to statement label T.	
ifls T	If Real(x) less than zero, then branch to statement label T.	
ifne T	If Real(x) is not equal to zero, then branch to statement label T.	
imag	Extract imaginary part. Set real part to zero.	Scalar mode: x = jIm(x) Mixed, vec: X = jIm(X)
impw	Generate impulse weighting.	Force window between markers 1 and 2. Response starts at marker 3 or 4, whichever is less, and decays to 1/e in |marker 4 - marker 3|.

Entry/ Mnemonic	Description	Notes
ina T	Input ASCII Record (See mini-manual part XIV for definitions of buffer abbreviations.)	T: aver, text, prog, x, y, z, t, aux, disp, meas, 1, 2, 3, 4.
inb T	Input binary record.	T: same as above
iwgt	Arbitrary function becomes impulse weighting function.	X (baseband time) becomes impulse (force) weighting function.
keyl TTT	Label soft keys.	T is a string of ASCII characters, 7 characters
N line	Same as LINE key.	N is a menu line number
lng	Natural log.	Scalar mode: x = ln(x) Mixed, vec: X = ln(X)
log	Common (base 10) log.	Scalar mode: x = log(x) Mixed, vec: X = log(X)
mag	Magnitude	Scalar mode: x = mag(x) Mixed, vec: X = mag(X)
max	Maximum	Scalar mode: if Re(y) > Re(x), then y goes into x. mixed, vec: if Re(Y) > Re(X), then Y goes into X, point by point.
maxm	Maximum magnitude.	Scalar mode: if mag(y) > mag(x), then y goes into X. mixed, vec: if mag(Y) > mag(X), then Y goes into X, point by point.
mclf	Menu cursor left.	
mcrt	Menu cursor right.	
min	Minimum	Opposite of "max".
minm	Minimum magnitude.	Opposite of "maxm".

Entry/ Mnemonic	Description	Note
mins	Minus (subtraction).	Scalar: x = y - x Mixed: X = X - x Vector: X = Y - X
nbw	Noise band width to x.	
ntc	Generate notch function, see "box". X = 1-box	
outa T	Output ASCII record.	T: see "ina"
outb T	Output binary record.	T: see "inb"
outc T	Output cursor format (scaled, decimal).	T: x, y, z or t
page	Set display to page number in Real(x). Possible pages are: 1 - Measurement menu 2 - Display menu 3 - Aux menu 4 - Single display 5 - Dual split 6 - Dual overlap 7 - Dual Nyquist	 8 - Quad split 9 - Quad overlap 10- Quad Nyquist, split 11- Quad Nyquist, overlap 12- Text/sigle 13- Text 0 - Update current display
peak	Returns value and position of peak magnitude of X between limits specified by x and y. Replaces x and y.	Parameters: x and y specify limits in Hz or seconds. Results: horizontal position of peak goes into x, mag of peak goes into y.
phse	Divide complex number by its magnitude. Result has phase of original function with magnitude equal to 1.	Scalar: x = x/mag(x) Mixed, vec: X=X/mag(X)
pls	Plus (addition)	Scalar mode: x = x+y Mixed mode: X = X+x Vector mode: X = X+Y
prmt T	Places text T on view line.	T: anything

Entry/ Mnemonic	Description	Notes
puty	Move data from y to location in X specified by x (cell #).	
pwr	10 raised to the power of X.	Scalar: $x = 10^x$ Mixed, vec: $X=10^X$
ramp	Generate ramp. Cursor position of each cell in Hz or seconds becomes that cell's value.	X = ramp
real	Extract real part. Set imaginary part to zero.	Scalar mode: Im(x)=0 Mixed, vec: Im(X)=0
recl T	Recall a scalar from one of the 52 scalar registers. Note: Post-increment and postdecrement of the real part of a recalled register may be achieved by adding a + or - sign after the argument, for example, recl a+ or recl a-.	T: a..z, A..Z. Note: A through L, Y & Z are used by the 804A. B-K hold marker positions, amplitudes. (Amplitudes generated by marker list.) A holds # of markers. L holds error code, see "serr".
recp	Reciprocal	Scalar mode: x = 1/x Mixed, vec: X = 1/X
ret	Return from subroutine.	
rev	Annotate program revision in Entry field.	Re(x) = rev Im(X) = serial number
rma	Return mean absolute value. Sum of amplitudes which should already be power.	x, y specify limits in Hz or seconds.
rms	Square root of sum of amplitude squared points between real(x) and real(y).	x, y specify limits in Hz or seconds.

Entry/ Mnemonic	Description	Note
roll	Roll down stack.	See Chapter 7.
run	Begin program execution at label in Entry field or at first line if Entry field is blank.	Not programmable.
rwgi	Recall impulse weighting.	Force weighting.
rwgr	Recall response weighting.	Response weighting
rwgt	Arbitrary function becomes response weighting.	X (baseband time) becomes response weighting function.
scel	Set cursor to cell #.	Cell # in x.
scur	Set cursor to position.	Position in x.
send T	Send string to remote control device.	T: anything.
serr	Errors are "soft". Soft errors generate an error code which is placed in the real part of register L, and do not stop SAMPL program execution. Due to the large number of possible error codes, they are printed once following the alphabetic listing of extended commands.	
sin	Sine	Scalar mode: x = sin(x) Mixed, vec: X = sin(X)
smix	Set mixed mode.	
soft TTT	Wait for soft key selection.	T: labels separated by spaces. Pressing first soft key causes program to continue at first label, etc.

Entry/ Mnemonic	Description	Notes
sqr	Square	Scalar mode: x = x * x Mixed, vec: X = X * X
sqrt	Square root.	Scalar mode: x = root(x) Mixed, vec: X = root(X)
sscl	Set scalar mode.	
stop	Stop program execution (can be used to wait for CONT to be pressed).	
stor T	Store x in scalar register.	T: a thru z or A thru Z. See "recl"
svec	Set vector mode.	
thrs	Threshold function.	If mag(X) is greater than or equal to mag(Y) then 1 goes into X else 0 goes into X at each point.
tims	Multiply	Scalar mode: x = x * y Mixed mode: X = X * x Vector mode: X = X * Y
tjw	Multiply by jω (differentiation)	Scalar mode: x = x * j where ω = ω at cursor vector, mixed: X = X * jω
tspu	Time span to x.	x: Time span in seconds
wait T	Wait for event.	T: avon, avof, nrt, key, arm, hold
?	Recalls "help" info for the command concerning the current position of the menu cursor. See "help".	Aux: "help" info goes to aux text area and is displayed.

A. Errors after "serr"

The following error codes may appear in register L if an error occurs following the "serr" command. Errors 01 through 70 are from the 804A program. Errors 128 through 168 are from the 804A's operating system.

Internal Error name	Code in Reg. L	Explanation
MSGNUL	00	NO ERROR OR WARNING - BLANK MESSAGE,NORMAL VIDE
ERNOR	01	NUMBER OUT OF RANGE
ERIFF	02	INPUT FIELD FULL
ERNRE	03	NO ROOM FOR EXPONENT
ERICH	04	INVALID CHARACTER IN INPUT
WR4CH	05	ONLY FIRST 4 CHARACTERS USED
WRCIG	06	INVALID COMMAND - IGNORED
ERMNF	07	PGM EXECUTION - MNEMONIC NOT FOUND
ERPRG	08	PGM EXECUTION - PARAMETER RANGE ERROR
ERPBF	09	BUFFERS ARE FULL
NMENB	10	ENTER NUMBER
NMEUN	11	ENTER UNITS
EREOF	12	ERROR END OF FILE
ERHLL	13	CANNOT SET MAX <= MIN
ERLGH	14	CANNOT SET MIN >= MAX
NMENS	15	ENTER NUMBER OR SELECTION
ERMKU	16	MARKER(S) UNDEFINED
EROV1	17	OVERLOAD ON CHAN A
EROV2	18	OVERLOAD ON CHAN B
EROV4	19	OVERLOAD ON CHAN C
EROV8	20	OVERLOAD ON CHAN D
EROV3	21	OVERLOAD ON CHAN A,B
EROV5	22	OVERLOAD ON CHAN A,C
EROV9	23	OVERLOAD ON CHAN A,D
EROV6	24	OVERLOAD ON CHAN B,C
EROVA	25	OVERLOAD ON CHAN B,D
EROVC	26	OVERLOAD ON CHAN C,D
EROV7	27	OVERLOAD ON CHAN A,B,C
EROVB	28	OVERLOAD ON CHAN A,B,D
EROVD	29	OVERLOAD ON CHAN A,C,D
EROVE	30	OVERLOAD ON CHAN B,C,D
EROVF	31	OVERLOAD ON CHAN A,B,C,D
NMAVC	32	AVERAGING COMPLETED
WRDFA	33	DISPLAY FUNC ADJUSTED
ERNNH	34	CANNOT SET MIN NEGATIVE
WRHFR	35	HORIZONTAL FORMAT RESET
ERMFZ	36	MEMORY IS FROZEN
ERVDD	37	ONLY ON DATA DISPLAY
WRHSA	38	HORIZ SCALE ADJUSTED
WRHSW	39	HORIZ SCALE TOO WIDE
ERSBD	40	TOO SMALL FOR SB DELTA
ERFCN	41	FREQ CURSOR MUST BE +

Internal Error name	Code in Reg. L	Explanations
ERCEB	42	CANNOT EDIT THIS BUFFER
ERKNU	43	KEY NOT USED IN THIS MODE
ERCTM	44	CANNOT FROM THIS MODE --
ERLIV	45	INVALID LABEL
ERLNF	46	LABEL NOT FOUND
EREMD	47	EXCEEDED MAX SUBR DEPTH
ERRLZ	48	CANNOT RTN FROM LEVEL 0
ERMNS	49	MEMORIES NOT SAME TYPE
WRHSU	50	HORIZ SCALES UNEQUAL
ERFNA	51	FUNCTION NOT AVAILABLE
ERITC	52	ILLEGAL TEMPLATE
NMDEL	53	DO YOU WISH TO DELETE:
ERNFM	54	NO FILES MATCH TEMPLATE
ERNUF	55	NOT A USER FILE
ERNBT	56	MEMORY NOT BASE BD TIME
ERBOL	57	RC BUFFER OVERFLOW IN LOCAL
ERCHE	58	IEEE488 CHIP ERROR
ERURC	59	UNRECOGNIZED COMMAND
EROCL	60	ONLY 1 CURSOR ALLOWED
ERDCI	61	DATA CURSOR INACTIVE
ERAMU	62	ALL MARKERS USED
ERNFF	63	NO FFT ON FREQ DATA
ERNIT	64	NO IFFT ON TIME DATA
ERPSR	65	PLOT SCALE SET TO FULL
ERZTA	66	CANNOT ZOOM AND TIME AVG
WRSCD	67	SCALING COMPLETED
ERCAF	68	CANNOT WHILE AVGING FREQ
NMEFN	69	ENTER FUNCTION NAME
WRSSC	70	VERY SLOW SAMPLING CLOCK
FEEXCL	128	NOT EXCLUSIVE USE
FEINVC	129	INVALID COMMAND
FEACTV	130	ACTIVE WHEN SHOULD BE INACTIVE
FENACT	131	INACTIVE WHEN SHOULD BE ACTIVE
FEDDEV	132	MUST BE DISK DEVICE
FEEXTS	133	FILE ALREADY EXISTS
FENOTE	134	FILE DOES NOT EXIST
FENSEK	135	NO SEEK ALLOWED
FEINVO	136	INVALID OPTIONS
FEPRIV	137	NOT PRIVLEGED TO CHANGE PERM OR DO DIRECT DISK I/O
FESEKE	138	SEEK PAST END OF FILE
FERDPR	139	READ PROTECTED
FEWRPR	140	WRITE PROTECTED
FEINVB	141	INVALID BLOCK POSITION

Internal Error name	Code in Reg. L	Explanations
FEEOF	142	END OF FILE ERROR
FEEXCT	143	ATEMPT TO EXTEND CONTIG FILE
FENODV	144	NO SUCH DEVICE
FESPAC	145	INSUFFICIENT DISK SPACE
FEBIGB	146	ATTEMPT TO ALLOCATE > 255 SECTORS
FENOCT	147	NOT ENOUGH CONTIG SECTORS
FENOBF	148	NO MORE BUFFERS
FELINK	149	LINKED LIST ERROR
FEINVL	150	INVALID BLOCK LENGTH
FEHWWP	151	HARDWARE WRITE PROTECT
FEHARD	152	HARD DISK ERROR
FEUNKE	153	UNKNOWN DISK ERROR
FERETR	154	TOO MANY RETRIES ON DISK
FERCLE	155	RE-CALIBRATE ERROR
FENORD	156	NO READY FROM DISK
FEMXCY	157	ATTEMPT TO SEEK PAST END OF DISK
FENCUR	158	NOT CURSOR POSITIONABLE
FENATR	159	NOT ATTRIBUTE SETTABLE
FEIVCU	160	INVALID CURSOR POSITION
FEINVA	161	INVALID ATTRIBUTE OR ADDRESS
FENOTI	162	DEVICE NOT INITIALIZED
FEOPND	163	FILES ARE OPENED ON DEVICE AT RELEASE TIME
FEINVN	164	INVALID NAME
FENDDV	165	MUST NOT BE A DISK DEVICE
FENCTL	166	NOT 488 CONTROL DEVICE
FEBUFA	167	488 BUFFER IS ACTIVE - CAN'T CHANGE MODE
FEBOVR	168	CIRCULAR BUFFER OVERFLOW

CHAPTER 7

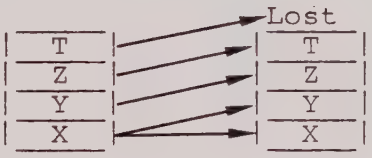
EXTENDED COMMANDS

7-1. Math Mode Control

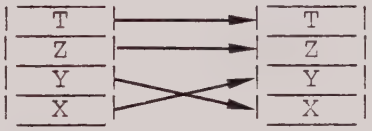
sscl : Set Scalar mode
svec : Set Vector mode
smix : Set Mixed mode (same as Vector except except for +, -, *,)

7-2. Stack Control

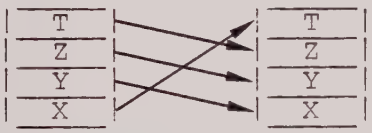
ent : Enter



exc : Exchange X & Y



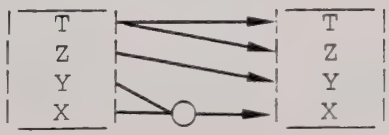
roll : Roll Stack



7-3. Basic Math Operations

pls :	Addition	Scalar	y+x→x
		Vector	Y+X→X
		Mixed	X+x→X
mins :	Subtraction	Scalar	y-x→x
		Vector	Y-X→X
		Mixed	X-x→X
tims :	Multiplication	Scalar	y*x→x
		Vector	Y*X→X
		Mixed	X*x→X
div :	Division	Scalar	y÷x→x
		Vector	Y÷X→X
		Mixed	X÷x→X

Stack movement for Scalar and Vector (no stack movement for Mixed)



7-4. Extended Math Operations

conj	:	Complex Conjugate	$X \rightarrow X$
recp	:	Reciprocal	$1/X \rightarrow X$
sqr	:	Square	$X*X \rightarrow X$
sqrt	:	Square root	$\sqrt{X} \rightarrow X$
tjw	:	Times jω (pseudo differentiation)	$X*j\omega \rightarrow X$
djw	:	Divide by jω (pseudo integration)	$X/j\omega \rightarrow X$

7-5. Transcendental Functions

sin	:	Sine function	$\sin(X) \rightarrow X$
cos	:	Cosine function	$\cos(X) \rightarrow X$
exp	:	Exponential	$e^X \rightarrow X$
lng	:	Natural logarithm	$\log_e(X) \rightarrow X$
pwr	:	Power of 10	$10^X \rightarrow X$
log	:	Logarithm base 10	$\log_{10}(X) \rightarrow X$

7-6. Format Conversion

mag	:	Magnitude	$\|X\| \rightarrow X$
real	:	Real part	$Re(X) \rightarrow X$
imag	:	Imaginary part	$jIm(X) \rightarrow X$
phse	:	Phase extraction	$X/\|X\| \rightarrow X$

7-7. Frequency/Time Transformations

fft : Fast Fourier Transform of $\underline{X} \rightarrow \underline{X}$
(X must be Time data).

ifft : Inverse Fast Fourier Transform of $\underline{X} \rightarrow \underline{X}$
(X must be Frequency data).

7-8. Vector Generation

cnst	:	Generate constant	$\underline{x} \rightarrow \underline{X}$
ramp	:	Generate ramp	$\underline{h} \rightarrow \underline{X}$ (1)
box	:	Generate box function	if $P_1 \leq h \leq P_2$, then $1,0 \rightarrow \underline{X}(1)\&(2)$ else $0,0 \rightarrow \underline{X}$
ntc	:	Generate notch function	if $P_1 \leq h \leq P_2$, then $0,0 \rightarrow \underline{X}$ else $1,0 \rightarrow \underline{X}(1)\&(2)$

---

(1) h is the cursor position in X in Hz or sec.

(2) P_1 is the minimum of Re(x) and Re(y).
P_2 is the maximum of Re(x) and Re(y).

7-9. Comparison and Peak

max :	Find maximum real part	If $Re(X) \geq Re(Y)$, then $X \rightarrow X$ else $Y \rightarrow X$
maxm :	Find maximum magnitude	If $\|X\| \geq \|Y\|$, then $X \rightarrow X$ else $Y \rightarrow X$
min :	Find minimum real part	If $Re(X) \leq Re(Y)$, then $X \rightarrow X$ else $Y \rightarrow X$
minm :	Find minimum magnitude	If $\|X\| \leq \|Y\|$, then $X \rightarrow X$ else $Y \rightarrow X$
thrs :	Threshold function	If $\|X\| \geq \|Y\|$, then $1,0 \rightarrow X$ else $0,0 \rightarrow X$
peak :	Peak pick function	Search $\underline{X}$ in region $P_1 \leq h \leq P_2$; h of maximum magnitude -> $\underline{y}$ X element of maximum magnitude $\rightarrow x^{(1)\&(2)}$

7-10. Cursor Control

dcof : Data cursor off
dcon : Data cursor on
dclf : Data cursor left (one step)
dcrt : Data cursor right (one step)
scur : Set cursor location to $\underline{x}$
scel : Set cursor cell number to $\underline{x}$

7-11. Access Parameters of Data Memory Associated with the Active Cursor

curh : Cursor horizontal position to $\underline{x}$
curv : Cursor vertical reading to $\underline{x}$
delf : Frequency resolution to $\underline{x}$
delt : Time resolution to $\underline{x}$
fspa : Frequency span to $\underline{x}$
tspa : Time span to $\underline{x}$
nbw : Noise bandwidth to $\underline{x}$

---

(1) is the cursor position in $\underline{X}$ in Hz or sec.

(2) P_1 is the minimum of $Re(\underline{x})$ and $Re(\underline{y})$.
P_2 is the maximum of $Re(\underline{x})$ and $Re(\underline{y})$.

7-12. Weighting Function Generation and Recall

impw : Generate impulse weighting functions based on markers 1 to 4.
iwgt : $\underline{X}$ (scaled) to the input (Channel A) weighting function.
rwgt : $\underline{X}$ (scaled) to the response (Channels B, C, and D) weighting function.
rwgi : Recall input weighting function to X.
rwgr : Recall response weighting function to X.

7-13. Conditional, Branching, and Program Control

goto <label> : Go to statement at <label>.
gosb <label> : Call subroutine at <label>.
ret : Return from subroutine.
ifeq <label> : Go to statement at <label> if $Re(\underline{x}) = 0$.
ifge <label> : Go to statement at <label> if $Re(\underline{x}) \geq 0$.
ifgt <label> : Go to statement at <label> if $Re(\underline{x}) > 0$.
ifle <label> : Go to statement at <label> if $Re(\underline{x}) \leq 0$.
ifls <label> : Go to statement at <label> if $Re(\underline{x}) < 0$.
ifne <label> : Go to statement at <label> if $Re(\underline{x}) \neq 0$.
run : Run a program starting at the label in the entry field or at the first statement if entry field is blank(3).
cont : Continue a stopped program(3).
stop : Stop program execution.
soft <label> <label> <label> : Wait for a SOFT key to be pressed, and then go to the statement at the <label> corresponding to that SOFT key.

---

(3) Cannot be used in a program.

7-14. I/O

```
ina  <buf>     :  Input to a buffer in ASCII coded
                  hexadecimal format
inb  <buf>     :  Input to a buffer in binary format
outa <buf>     :  Output from a buffer in ASCII
                  coded hexadecimal format
outb <buf>     :  Output from a buffer in binary format
outc <buf>     :  Output from a buffer in character
                  format (only buffers x, y, z, or t allowed)
send <text>    :  Output the text to the current I/O device
copy <file1>   :  Copy from <file1> to <file2>
     <file2>
```

Possible Buffers (<buf>)

```
x            :  X data memory
y            :  Y data memory
z            :  Z data memory
t            :  T data memory
meas         :  Measurement setup states
disp         :  Display setup states
aux          :  Auxiliary setup states
aver         :  Averager memory
text         :  Text area
prog         :  Program area
aux a        :  Auxiliary text area
1            :  Trace 1
2            :  Trace 2
3            :  Trace 3
4            :  Trace 4
V1           :  Vector memory 1
V2           :  Vector memory 2
V3           :  Vector memory 3
V4           :  Vector memory 4
```

7-15. Miscellaneous Commands

```
stor <reg>     :  Store x in register <reg>;
                  reg A...Z, a...z.
recl <reg>     :  Recall register to x.
page           :  Select page number x.
prmt <text>    :  Put <text> in the view line
                  as a operator prompt.
put            :  Move y to the location in X
                  at location of x.
keyl <text>    :  Label the SOFT keys with the text
                  (seven characters per label).
dely           :  Delay program execution for x miliseconds.
```

rev : Put program revision and hardware serial number into the entry field.

rms : Calculate the RMS (root mean squared) value of the data in X with $P_1 \leq h \leq P_2$ and put it in $\underline{x}$(1)&(2)

rma : Calculate the RMA (mean absolute value of the data in X with $P_1 \leq h \leq P_2$, and put it in $\underline{x}$ (1)&(2)

wait <event> : Wait for an event. Events are

- arm : Transient capture armed
- hold : Transient capture in hold
- avon : Averager on
- avof : Averager off
- key : Any key pressed
- nrt : Non-redundant data gathered
- help : Display a lsit of all commands or
- <command> information on one command
- clr<area>: Clear an area. Areas are
- text : Text area
- prog : Program area
- aver : Averager buffer
- auxa : Auxiliary text area
- x : $\underline{X}$ if vector mode
  $\underline{x}$ if scalar mode
  $\underline{X}$ and $\underline{x}$ if mixed mode
- stk : $\underline{X}$, $\underline{Y}$, $\underline{Z}$, $\underline{T}$ if vector mode
  $\underline{x}$, $\underline{y}$, $\underline{z}$, $\underline{t}$ if scalar mode
  $\underline{X}$, $\underline{Y}$, $\underline{Z}$, $\underline{T}$, $\underline{x}$, $\underline{y}$, $\underline{z}$, $\underline{t}$ if mixed mode
- data : Turn on display of last viewed data and graticule
- datm : Display data and time and month, day, hour -> $\underline{x}$
- clbl : Set or display the current cursor and data labels

bus (opt) : Perform control functions on the 488 bus

---

(1) h is the cursor position in $\underline{X}$ in Hz or sec.

(2) P_1 is the maximum of Re($\underline{x}$) and Re($\underline{y}$).
P_2 is the maximum of Re($\underline{x}$) and Re($\underline{y}$).

Options are

rtl	:	Return to local
ifc	:	Assert interface clear
rem	:	Assert Remote enable
loc	:	Negate Remote enable
llo	:	Send local lockout message
gll	:	Send go to local message
sdc	:	Send selective device clear message
dcl	:	Send device clear message
get	:	Send group execute trigger message

line	:	Same as |LINE| key
mclf	:	Menu cursor left
mcrt	:	Menu cursor right